Engineering a Safer World

Systems Thinking Applied to Safety

(This is a draft. It's complete but still is undergoing professional editing. Expected publication date by MIT Press is Fall, 2011.)

Nancy G. Leveson

Aeronautics and Astronautics and
Engineering Systems Division

Massachusetts Institute of Technology

We pretend that technology, our technology, is something of a life force, a will, and a thrust of its own, on which we can blame all, with which we can explain all, and in the end by means of which we can excuse ourselves.

— T. Cuyler Young
Man in Nature

DEDICATION: To all the great engineers who taught me system safety engineering, particularly Grady Lee who believed in me. Also to those who created the early foundations for applying systems thinking to safety, including C.O. Miller and the other American aerospace engineers who created System Safety in the U.S. as well as Jens Rasmussen's pioneering work in Europe.

Preface

I began my adventure in system safety after completing graduate studies in computer science and joining the faculty of a computer science department. In the first week at my new job, I received a phone call from Marion Moon, a system safety engineer at what was then Ground Systems Division of Hughes Aircraft Company. Apparently he had been passed between several faculty members—I was his last hope. He told me about a new problem they were struggling with on a torpedo project, something he called "software safety." I told him I didn't know anything about it, that I worked in a completely unrelated field, but I was willing to look into it. That began what has been a thirty year search for a solution to his problem and to the more general problem of how to build safer systems.

Around the year 2000, I became very discouraged. Although many bright people had been working on the problem of safety for a long time, progress seemed to be stalled. Engineers were diligently performing safety analyses that did not seem to have much impact on accidents. The reason for the lack of progress, I decided, was that the technical foundations and assumptions on which traditional safety engineering efforts are based are inadequate for the complex systems we are building today.

The world of engineering has experienced a technological revolution while the basic engineering techniques applied in safety and reliability engineering, such as Fault Tree Analysis (FTA) and Failure Modes and Effects Analysis (FMEA), have changed very little. Few systems are built without digital components, which operate very differently than the purely analog systems they replace. At the same time, the complexity of our systems and the world in which they operate has also increased enormously. The old safety engineering techniques, which were based on a much simpler, analog world, are diminishing in their effectiveness as the cause of accidents changes.

For twenty years I watched engineers in industry struggling to apply the old techniques to new software-intensive systems—expending much energy and having little success. At the same time, engineers can no longer focus only on technical issues and ignore the social, managerial, and even political factors that impact safety if we are to significantly reduce losses. I decided to search for something new. This book describes the results of that search and the new model of accident causation and system safety techniques that resulted.

The solution, I believe, lies in creating approaches to safety based on modern systems thinking and systems theory. While these approaches may seem new or paradigm changing, they are rooted in system engineering ideas developed after World War II. They also build on the unique approach to engineering for safety, called System Safety, that was pioneered in the 1950s by aerospace engineers such as C.O. Miller, Jerome Lederer, Willie Hammer, and many others. This systems approach to safety was created originally to cope with the increased level of complexity in aerospace systems, particularly military aircraft and ballistic missile systems. Many of these ideas have been lost over the years or have been displaced by the influence of more mainstream engineering practices, particularly reliability engineering.

This book returns to these early ideas and updates them for today's technology. It also builds on the pioneering work in Europe of Jens Rasmussen and his followers in applying systems thinking to safety and human factors engineering.

Our experience to date is that the new approach described in this book is more effective, less expensive, and easier to use than current techniques. I hope you find it useful.

Relationship to *Safeware*

My first book, *Safeware*, presents a broad overview of what is known and practiced in System Safety today and provides a reference for understanding the state of the art. To avoid redundancy, information about basic concepts in safety engineering that appear in *Safeware* are not, in general, repeated. To make this book coherent in itself, however, there is some repetition, particularly on topics for which my understanding has advanced since writing *Safeware*.

Audience

This book is written for the sophisticated practitioner rather than the academic researcher or the general public. Therefore, although references are provided, an attempt is not made to cite or describe everything ever written on the topics or to provide a scholarly analysis of the state of research in this area. The goal is to provide engineers and others concerned about safety with some tools they can use when attempting to reduce accidents and make systems and sophisticated products safer.

It is also written for those who are not safety engineers and those who are not even engineers. The approach described can be applied to any complex, sociotechnical system such as health care and even finance. This book shows how to "re-engineer" your system to improve safety and better manage risk. If preventing potential losses in your field is important, then the answer to your problems may lie in this book.

Contents

The basic premise underlying this new approach to safety is that traditional models of causality need to be extended to handle today's engineered systems. The most common accident causality models assume that accidents are caused by component failure and that making system components highly reliable or planning for their failure will prevent accidents. While this assumption is true in the relatively simple electro-mechanical systems of the past, it is no longer true for the types of complex sociotechnical systems we are building today. A new, extended model of accident causation is needed to underlie more effective engineering approaches to improving safety and better managing risk.

The book is divided into three sections. The first part explains why a new approach is needed, including the limitations of traditional accident models, the goals for a new model, and the fundamental ideas in system theory upon which the new model is based. Part 2 presents the new, extended causality model. The final part shows how the new model can be used to create new techniques for system safety engineering, including accident investigation and analysis, hazard analysis, design for safety, operations, and management.

This book has been a long time in preparation because I wanted to try the new techniques myself on real systems to make sure they work and are effective. In order not to delay publication further, I will create exercises, more examples, and other teaching and learning aids and provide them for download from a website in the future.

Chapters 6 through 10 on system safety engineering and hazard analysis are purposely written to be stand-alone and therefore usable in undergraduate and graduate system engineering classes where safety is just one part of the class contents and the practical design aspects of safety are the most relevant.

Acknowledgements

The research that resulted in this book was partially supported by numerous research grants over many years from NSF and NASA. David Eckhardt at the NASA Langley Research Center provided the early funding that got this work started.

I also am indebted to all my students and colleagues who have helped develop these ideas over the years. There are too many to list, but I have tried to give them credit throughout the book for the ideas they came up with or we worked on together. I apologize in advance if I have inadvertently not given credit where it is due. My students, colleagues, and I engage in continual discussions and sharing of ideas—it is sometimes difficult to determine where the ideas originated. Usually the creation involves a process where we each build on what the other has done. Determining who is responsible for what becomes impossible. Needless to say, they provided invaluable input and contributed greatly to my thinking.

I am particularly indebted to the students who were at MIT while I was writing this book and played an important role in developing the ideas: Nicolas Dulac, Margaret Stringfellow, Brandon Owens, Matthieu Couturier, and John Thomas. Several of them assisted with the examples used in this book.

Other former students who provided important input to the ideas in this book are Matt Jaffe, Elwin Ong, Natasha Neogi, Karen Marais, Kathryn Weiss, David Zipkin, Stephen Friedenthal, Michael Moore, Mirna Daouk, John Stealey, Stephanie Chiesi, Brian Wong, Mal Atherton, Shuichiro Daniel Ota, and Polly Allen.

Colleagues who provided assistance and input include Sidney Dekker, John Carroll, Joel Cutcher-Gershenfeld, Joseph Sussman, Betty Barrett, Ed Bachelder, Margaret-Anne Storey, Meghan Dierks, and Stan Finkelstein.

Contents

Part I

Foundations

Chapter 1

Why Do We Need Something Different?

This book presents a new approach to building safer systems that departs in important ways from traditional safety engineering. While the traditional approaches worked well for the simpler systems of the past for which they were devised, significant changes have occurred in the types of systems we are attempting to build today and the context in which they are being built. These changes are stretching the limits of safety engineering:

- **Fast pace of technological change**: While learning from past accidents is still an important part of safety engineering, lessons learned over centuries about designing to prevent accidents may be lost or become ineffective when older technologies are replaced with new ones. Technology is changing much faster than our engineering techniques are responding to these changes. New technology introduces unknowns into our systems and creates new paths to losses.

- **Reduced ability to learn from experience**: At the same time that the development of new technology has sprinted forward, the time to market for new products has greatly decreased and strong pressures exist to decrease this time even further. The average time to translate a basic technical discovery into a commercial product in the early part of this century was 30 years. Today our technologies get to market in two to three years and may be obsolete in five. We no longer have the luxury of carefully testing systems and designs to understand all the potential behaviors and risks before commercial or scientific use.

- **Changing nature of accidents**: As our technology and society change, so do the causes of accidents. System engineering and system safety engineering techniques have not kept up with the rapid pace of technological innovation. Digital technology, in particular, has created a quiet revolution in most fields of engineering. Many of the approaches to prevent accidents that worked on electromechanical components—such as replication of components to protect against individual component failure—are ineffective in controlling accidents that arise from the use of digital systems and software.

- **New types of hazards**: Advances in science and societal changes have created new hazards. For example, the public is increasingly being exposed to new man-made chemicals or toxins in our food and our environment. Large numbers of people may be harmed by unknown side effects of pharmaceutical products. Misuse or overuse of antibiotics has given rise to resistant microbes. The most common safety engineering strategies have limited impact on many of these new hazards.

- **Increasing complexity and coupling**: Complexity comes in many forms, most of which are increasing in the systems we are building. Examples include *interactive complexity* (related to interaction among system components), *dynamic complexity* (related to changes over time), *decompositional complexity* (where the structural decomposition is not consistent with the functional decomposition), and *nonlinear complexity* (where cause and effect are not related in a direct or obvious way). The operation of some systems is so complex that it defies the understanding of all but a few experts, and sometimes even they have incomplete information about the system's potential behavior. The problem is that we are attempting to build systems that are beyond our ability to intellectually manage: Increased complexity of all types makes it difficult for the designers to consider all the potential system states or for operators to handle all normal and abnormal situations and disturbances safely and effectively. In fact, complexity can be defined as intellectual unmanageability.

  This situation is not new. Throughout history, inventions and new technology have often gotten ahead of their scientific underpinnings and engineering knowledge, but the result has always been increased risk and accidents until science and engineering caught up.[1] We are now in the position of having to catch up with our technological advances by greatly increasing the power of current approaches to controlling risk and creating new improved risk management strategies.

- **Decreasing tolerance for single accidents**: The losses stemming from accidents is increasing with the cost and potential destructiveness of the systems we build. New scientific and technological discoveries have not only created new or increased hazards (such as radiation exposure and chemical pollution) but have also provided the means to harm increasing numbers of people as the scale of our systems increases and to impact future generations through environmental pollution and genetic damage. Financial losses and lost potential for scientific advances are also increasing in an age where, for example, a spacecraft may take ten years and up to a billion dollars to build but only a few minutes to lose. Financial system meltdowns can impact the world's economy in our increasingly connected and interdependent global economy. Learning from accidents or major losses (the *fly-fix-fly* approach to safety) needs to be supplemented with increasing emphasis on preventing the first one.

- **Difficulty in selecting priorities and making tradeoffs**: At the same time that potential losses from single accidents is increasing, companies are coping with aggressive and compet-

[1] As an example, consider the introduction of high-pressure steam engines in the first half of the nineteenth century, which transformed industry and transportation but resulted in frequent and disastrous explosions. While engineers quickly amassed scientific information about thermodynamics, the action of steam in the cylinder, the strength of materials in the engine and many other aspects of steam engine operation, there was little scientific understanding about the buildup of steam pressure in the boiler, the effect of corrosion and decay, and the causes of boiler explosions. High-pressure steam had made the current boiler design obsolete by producing excessive strain on the boilers and exposing weaknesses in the materials and construction. Attempts to add technological safety devices were unsuccessful because engineers did not fully understand what went on in steam boilers: It was not until well after the mid-century that the dynamics of steam generation was understood [28].

itive environments in which cost and productivity play a major role in short-term decision making. Government agencies must cope with budget limitations in an age of increasingly expensive technology. Pressures are great to take shortcuts and to place higher priority on cost and schedule risks than on safety. Decision makers need the information required to make these tough decisions.

- **More complex relationships between humans and automation**: Humans are increasingly sharing control of systems with automation and moving into positions of higher-level decision making with automation implementing the decisions. These changes are leading to new types of human error—such as various types of mode confusion—and a new distribution of human errors, for example, increasing errors of omission versus commission [181, 182]. Inadequate communication between humans and machines is becoming an increasingly important factor in accidents. Current approaches to safety engineering are unable to deal with these new types of errors.

  All human behavior is influenced by the context in which it occurs, and operators in high-tech systems are often at the mercy of the design of the automation they use or the social and organizational environment in which they work. Many recent accidents that have been blamed on operator error could more accurately be labeled as resulting from flaws in the environment in which they operate. New approaches to reducing accidents through improved design of the workplace and of automation are long overdue.

- **Changing regulatory and public views of safety**: In today's complex and interrelated societal structure, responsibility for safety is shifting from the individual to government. Individuals no longer have the ability to control the risks around them and are demanding that government assume greater responsibility for ensuring public safety through laws and various forms of oversight and regulation as companies struggle to balance safety risks with pressure to satisfy time-to-market and budgetary pressures. Ways to design more effective regulatory strategies without impeding economic goals are needed. The alternative is for individuals and groups to turn to the courts for protection, which has many potential downsides, such as stifling innovation through fear of lawsuits as well as unnecessarily increasing costs and decreasing access to products and services.

Incremental improvements in traditional safety engineering approaches over time have not resulted in significant improvement in our ability to engineer safer systems. A paradigm change may be needed to make the improvements needed for the types of systems and hazards we are dealing with today. This book shows how systems theory and systems thinking can be used to extend our understanding of accident causation and provide more powerful (and surprisingly less costly) new accident analysis and prevention techniques. It also allows a broader definition of safety and accidents that go beyond human death and injury and includes all types of major losses including equipment, mission, financial, and information.

Part I of this book presents the foundation for the new approach. The first step is to question the current assumptions and oversimplifications about the cause of accidents that no longer fit today's systems (if they ever did) and create new assumptions to guide future progress. The new, more realistic assumptions are used to create goals to reach for and criteria against which new approaches can be judged. Finally, the scientific and engineering foundations for a new approach are outlined.

Part II presents a new, more inclusive model of causality, followed by Part III, which describes how to take advantage of the expanded accident causality model to better manage safety in the twenty-first century.

Chapter 2

Questioning the Foundations of Traditional Safety Engineering

It's never what we don't know that stops us. It's what we do know that just ain't so.[1]

Paradigm changes necessarily start with questioning the basic assumptions underlying what we do today. Many beliefs about safety and why accidents occur have been widely accepted without question. This chapter examines and questions some of the most important assumptions about the cause of accidents and how to prevent them that "just ain't so." There is, of course, some truth in each of these assumptions, and many were true for the systems of the past. The real question is whether they still fit today's complex sociotechnical systems and what new assumptions need to be substituted or added.

2.1 Confusing Safety with Reliability

Assumption 1: Safety is increased by increasing system or component reliability. If components or systems do not fail, then accidents will not occur.

This assumption is one of the most pervasive in engineering and other fields. The problem is that it's not true. Safety and reliability are *different* properties. One does not imply nor require the other: A system can be reliable but unsafe. It can also be safe but unreliable. In some cases, these two properties even conflict, that is, making the system safer may decrease reliability and enhancing reliability may decrease safety. The confusion on this point is exemplified by the primary focus on failure events in most accident and incident analysis. Some researchers in organizational aspects of safety also make this mistake by suggesting that high *reliability* organizations will be safe [106, 174, 175, 205, 206].

Because this assumption about the equivalence between safety and reliability is so widely held, the distinction between these two properties needs to be carefully considered. First let's consider accidents where nothing fails.

[1] Attributed to Will Rogers (e.g., New York Times, 10/7/84, p. B4), Mark Twain, and Josh Billings (Oxford Dictionary of Quotations, 1979 p.49) among others.

Reliable but Unsafe: In complex systems, accidents often result from interactions among components that are all satisfying their individual requirements, that is, they have *not* failed. The loss of the Mars Polar Lander was attributed to noise (spurious signals) generated when the landing legs were deployed during the spacecraft's descent to the planet surface [94]. This noise was normal and expected and did not represent a failure in the landing leg system. The onboard software interpreted these signals as an indication that landing had occurred (which the software engineers were told such signals would indicate) and shut down the descent engines prematurely, causing the spacecraft to crash into the Mars surface. The landing legs and the software performed correctly (as specified in their requirements) and reliably, but the accident occurred because the system designers did not account for all the potential interactions between landing leg deployment and the descent engine control software.

The Mars Polar Lander loss is a *component interaction accident*. Such accidents arise in the interactions among system components (electromechanical, digital, human, and social) rather than in the failure of individual components. In contrast, the other main type of accident, a *component failure accident*, results from component failures, including the possibility of multiple and cascading failures. In component failure accidents, the failures are usually treated as random phenomena. In component interaction accidents, there may be no failures and the system design errors giving rise to unsafe behavior are not random events.

A *failure* in engineering can be defined as the non-performance or inability of a component (or system) to perform its intended function. Intended function (and thus failure) is defined with respect to the component's behavioral requirements. If the behavior of a component satisfies its specified requirements (such as turning off the descent engines when a signal from the landing legs is received), even though the requirements may include behavior that is undesirable from a larger system context, that component has *not* failed.

Component failure accidents have received the most attention in engineering, but component interaction accidents are becoming more common as the complexity of our system designs increases. In the past, our designs were more intellectually manageable and the potential interactions among components could be thoroughly planned, understood, anticipated, and guarded against [154]. In addition, thorough testing was possible and could be used to eliminate design errors before use. Modern, high-tech systems no longer have these properties and system design errors are increasingly the cause of major accidents, even when all the components have operated reliably—that is, the components have not failed.

Consider another example of a component interaction accident that occurred in a batch chemical reactor in England [102]. The design of this system is shown in Figure 2.1. The computer was responsible for controlling the flow of catalyst into the reactor and also the flow of water into the reflux condenser to cool off the reaction. Additionally, sensor inputs to the computer were supposed to warn of any problems in various parts of the plant. The programmers were told that if a fault occurred in the plant, they were to leave all controlled variables as they were and to sound an alarm.

On one occasion, the computer received a signal indicating a low oil level in a gearbox. The computer reacted as the requirements specified: It sounded an alarm and left everything as it was. By coincidence, a catalyst had just been added to the reactor, but the computer had only started to increase the cooling-water flow to the reflux condenser; the flow was therefore kept at a low rate. The reactor overheated, the relief valve lifted, and the content of the reactor was discharged into the atmosphere.

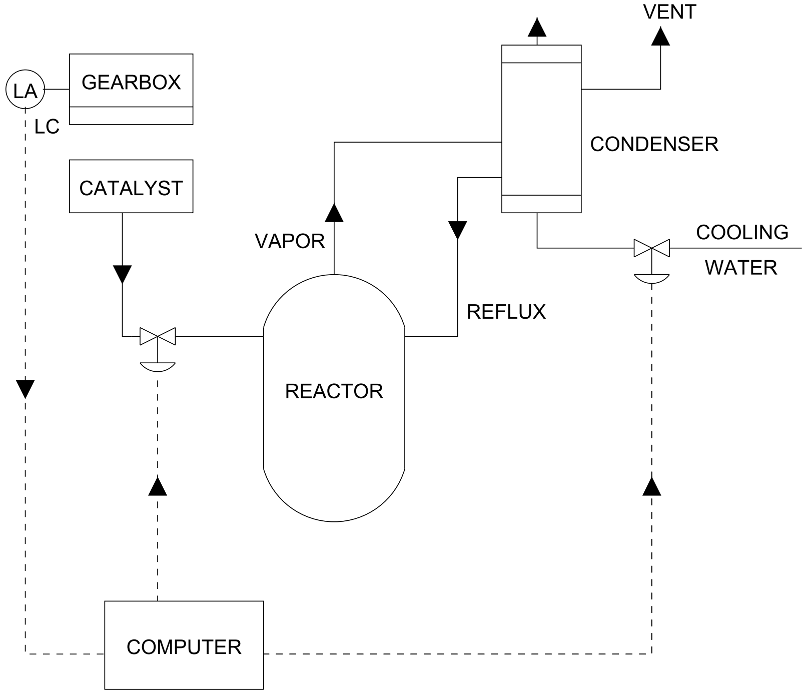


Figure 2.1: A chemical reactor design (adapted from Kletz [102, p.6]).

Note that there were no component failures involved in this accident: the individual components, including the software, worked as specified, but together they created a hazardous system state. The problem was in the overall system design. Merely increasing the reliability of the individual components or protecting against their failure would not have prevented this accident because none of the components failed. Prevention required identifying and eliminating or mitigating unsafe interactions among the system components. High component reliability does not prevent component interaction accidents.

Safe but Unreliable: Accidents like the Mars Polar Lander or the British batch chemical reactor losses, where the cause lies in dysfunctional interactions of non-failing, reliable components—i.e., the problem is in the overall system design—illustrate reliable components in an unsafe system. There can also be safe systems with unreliable components if the system is designed and operated so that component failures do not create hazardous system states. Design techniques to prevent accidents are described in Chapter 16 of *Safeware*. One obvious example is systems that are fail-safe, that is, they are designed to fail into a safe state.

For an example of behavior that is unreliable but safe, consider human operators. If operators do not follow the specified procedures, then they are not operating reliably. In some cases that can lead to an accident. In other cases, it may prevent an accident when the specified procedures turn out to be unsafe under the particular circumstances existing at that time. Examples abound of operators ignoring prescribed procedures in order to prevent an accident [114, 154]. At the same time, accidents have resulted precisely because the operators *did* follow the predetermined instructions provided to them in their training, such as at Three Mile Island [114]. When the results of deviating from procedures are positive, operators are lauded, but when the results are negative, they are punished for being "unreliable." In the successful case (deviating from specified procedures averts an accident), their behavior is unreliable but safe. It satisfies the behavioral safety constraints for the system, but not individual reliability requirements with respect to following specified procedures.

It may be helpful at this point to provide some additional definitions. *Reliability* in engineering is defined as the probability that something satisfies its specified behavioral requirements over time and under given conditions—that is, it does not fail [114]. Reliability is often quantified as *mean time between failure*. Every hardware component (and most humans) can be made to "break" or fail given some set of conditions or a long enough time. The limitations in time and operating conditions in the definition are required to differentiate between (1) unreliability under the assumed operating conditions and (2) situations where no component or component design could have continued to operate.

If a driver engages the brakes of a car too late to avoid hitting the car in front, we would not say that the brakes "failed" because they did not stop the car under circumstances for which they were not designed. The brakes, in this case, were *not* unreliable. They operated reliably but the requirements for safety went beyond the capabilities of the brake design. Failure and reliability are always related to requirements and assumed operating (environmental) conditions. If there are no requirements either specified or assumed, then there can be no failure as any behavior is acceptable and no unreliability.

Safety, in contrast, is defined as the absence of accidents, where an accident is an event involving an unplanned and unacceptable loss [114]. To increase safety, the focus should be on eliminating or preventing hazards, not eliminating failures. Making all the components highly reliable will not necessarily make the system safe.

Conflicts Between Safety and Reliability: At this point you may be convinced that reliable *components* are not enough for system safety. But surely, if the *system* as a whole is reliable it will be safe and vice versa, if the system is unreliable it will be unsafe. That is, reliability and safety are the same thing at the system level, aren't they? This common assumption is also untrue. A chemical plant may very reliably manufacture chemicals while occasionally (or even continually) releasing toxic materials into the surrounding environment. The plant is reliable but unsafe.

Not only are safety and reliability not the same thing, but they sometimes conflict: Increasing reliability may decrease safety and increasing safety may decrease reliability. Consider the following simple example in physical design. Increasing the working pressure to burst ratio (essentially the strength) of a tank will make the tank more reliable, that is, it will increase the mean time between failure. When a failure does occur, however, more serious damage may result because of the higher pressure at the time of the rupture.

Reliability and safety may also conflict in engineering design when a choice has to be made between retreating to a fail-safe state (and protecting people and property) versus attempting to continue to achieve the system objectives but with increased risk of an accident.

Understanding the conflicts between reliability and safety requires distinguishing between requirements and constraints. Requirements are derived from the mission or reason for the existence of the organization. The mission of the chemical plant is to produce chemicals. Constraints represent acceptable ways the system or organization can achieve the mission goals. Not exposing bystanders to toxins and not polluting the environment are constraints on the way the mission (producing chemicals) can be achieved.

While in some systems safety is part of the mission or reason for existence, such as air traffic control or healthcare, in others safety is not the mission but instead is a constraint on how the mission can be achieved. The best way to ensure the constraints are enforced in such a system may be not to build or operate the system at all. Not building a nuclear bomb is the surest protection against accidental detonation. We may be unwilling to make that compromise, but some compromise is almost always necessary: The most effective design protections (besides not building the bomb at all) against accidental detonation also decrease the likelihood of detonation when it is required.

Not only do safety constraints sometimes conflict with mission goals, but the safety requirements may even conflict among themselves. One safety constraint on an automated train door system, for example, is that the doors must not open unless the train is stopped and properly aligned with a station platform. Another safety constraint is that the doors must open anywhere for emergency evacuation. Resolving these conflicts is one of the important steps in safety and system engineering.

Even systems with mission goals that include assuring safety, such as air traffic control (ATC), usually have other conflicting goals. ATC systems commonly have the mission to both increase system throughput and ensure safety. One way to increase throughput is to decrease safety margins by operating aircraft closer together. Keeping the aircraft separated adequately to assure acceptable risk may decrease system throughput.

There are always multiple goals and constraints for any system—the challenge in engineering design and risk management is to identify and analyze the conflicts, to make appropriate tradeoffs among the conflicting requirements and constraints, and to find ways to increase system safety without decreasing system reliability.

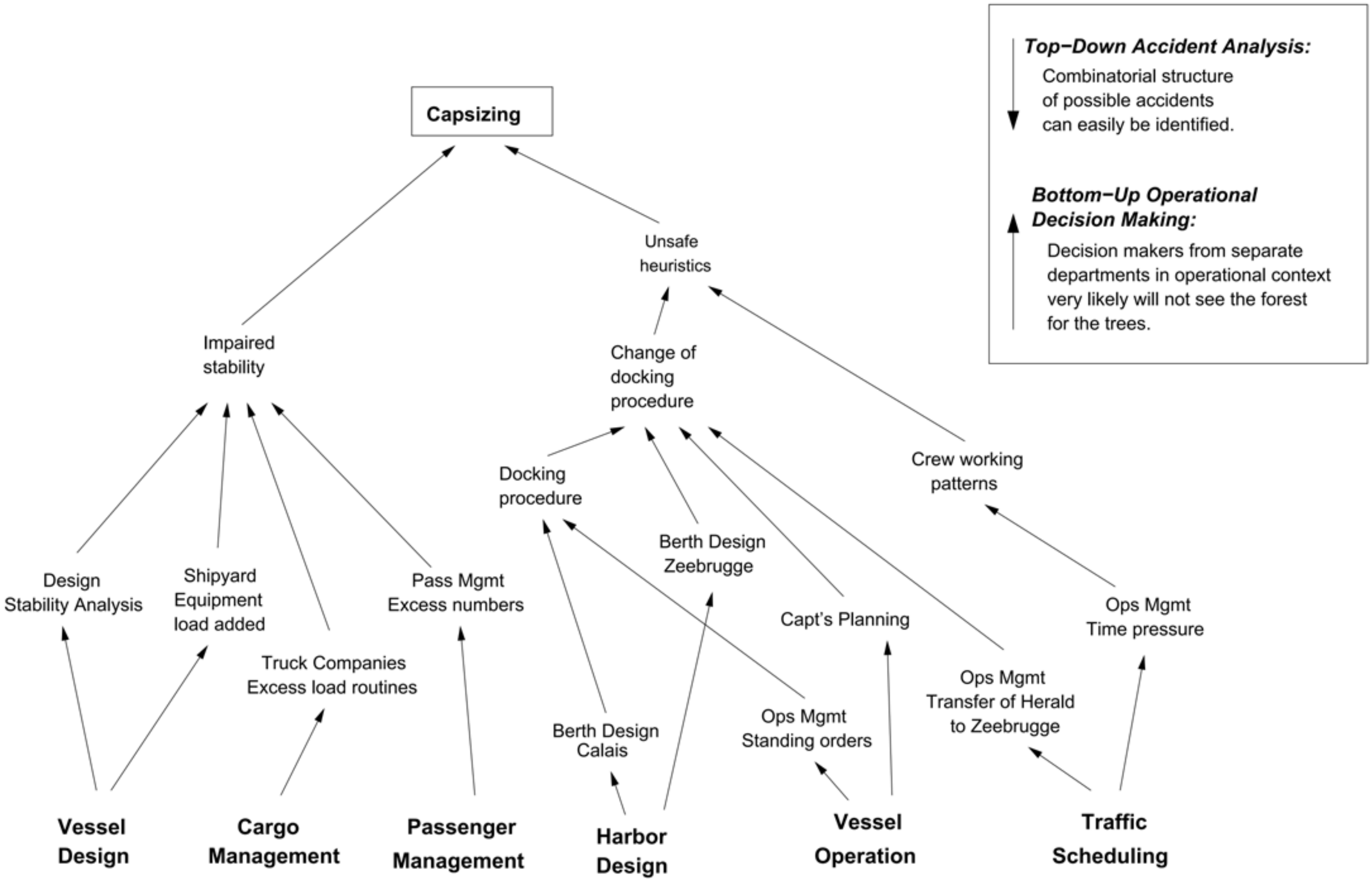

Top−Down Accident Analysis:
Combinatorial structure of possible accidents can easily be identified.
Bottom−Up Operational Decision Making:
Decision makers from separate departments in operational context very likely will not see the forest for the trees.
Capsizing
Unsafe heuristics
Impaired stability
Change of docking procedure
Crew working patterns
Docking procedure
Berth Design Zeebrugge
Design Stability Analysis
Shipyard Equipment load added
Pass Mgmt Excess numbers
Capt's Planning
Ops Mgmt Time pressure
Truck Companies Excess load routines
Ops Mgmt Transfer of Herald to Zeebrugge
Berth Design Calais
Ops Mgmt Standing orders
Vessel Design
Cargo Management
Passenger Management
Harbor Design
Vessel Operation
Traffic Scheduling

Figure 2.2: The complex interactions in the Zeebrugge accident (Adapted from Ramussen [166, p.188]).

Safety versus Reliability at the Organizational Level: So far the discussion has focused on safety versus reliability at the physical level. But what about the social and organizational levels above the physical system? Are safety and reliability the same here as implied by High Reliability Organization (HRO) advocates who suggest that High Reliability Organizations (HROs) will be safe? The answer, again, is no [123].

Figure 1 shows Rasmussen's analysis of the Zeebrugge ferry mishap [166]. Some background is necessary to understand the figure. On the day the ferry capsized, the *Herald of Free Enterprise* was working the route between Dover and the Belgium port of Bruges–Zeebrugge. This route was not her normal one and the linkspan[2] at Zeebrugge had not been designed specifically for the Spirit type of ships. The linkspan used spanned a single deck and so could not be used to load decks E and G simultaneously. The ramp could also not be raised high enough to meet the level of deck E due to the high spring tides at that time. This limitation was commonly known and was overcome by filling the forward ballast tanks to lower the ferry's bow in the water. The *Herald* was due to be modified during its refit later that year to overcome this limitation in the ship's design.

[2] A *linkspan* is a type of drawbridge used in moving vehicles on and off ferries or other vessels.

Before dropping moorings, it was normal practice for a member of the crew, the assistant bosun, to close the ferry doors. The first officer also remained on deck to ensure they were closed before returning to the wheelhouse. To keep on schedule, the first officer returned to the wheelhouse before the ship dropped its moorings (which was common practice), leaving the closing of the doors to the assistant bosun, who had taken a short break after cleaning the car deck upon arrival at Zeebrugge. He had returned to his cabin and was still asleep when the ship left the dock. The captain could only assume that the doors had been closed because he could not see them from the wheelhouse due to their construction, and there was no indicator light in the wheelhouse to show door position. Why nobody else closed the door is unexplained in the accident report.

Other factors also contributed to the loss. One was the depth of the water: if the ship's speed had been below 18 knots (33 km/h) and the ship had not been in shallow water, it was speculated in the accident report that the people on the car deck would probably have had time to notice the bow doors were open and close them [186]. But open bow doors were not alone enough to cause the final capsizing. A few years earlier one of the *Herald*'s sister ships sailed from Dover to Zeebrugge with the bow doors open and made it to her destination without incident.

Almost all ships are divided into watertight compartments below the waterline so that in the event of flooding, the water will be confined to one compartment, keeping the ship afloat. The *Herald*'s design had an open car deck with no dividers, allowing vehicles to drive in and out easily, but this design allowed water to flood the car deck. As the ferry turned, the water on the car deck moved to one side and the vessel capsized. One hundred and ninety three passengers and crew were killed.

In this accident, those making decisions about vessel design, harbor design, cargo management, passenger management, traffic scheduling, and vessel operation were unaware of the impact (side effects) of their decisions on the others and the overall impact on the process leading to the ferry accident. Each operated "reliably" in terms of making decisions based on the information they had.

Bottom-up decentralized decision making can lead—and has led—to major accidents in complex sociotechnical systems. Each local decision may be "correct" in the limited context in which it was made but lead to an accident when the independent decisions and organizational behaviors interact in dysfunctional ways.

Safety is a system property, not a component property, and must be controlled at the system level, not the component level. We return to this topic in Chapter 3.

Assumption 1 is clearly untrue. A new assumption needs to be substituted:

New Assumption 1: High reliability is neither necessary nor sufficient for safety.

Building safer systems requires going beyond the usual focus on component failure and reliability to focus on system hazards and eliminating or reducing their occurrence. This fact has important implications for analyzing and designing for safety. Bottom-up reliability engineering analysis techniques, such as FMEA, are not appropriate for safety analysis. Even top-down techniques, such as fault trees, if they focus on component failure, are not adequate. Something else is needed.

2.2 Modeling Accident Causation as Event Chains

Assumption 2. Accidents are caused by chains of directly related events. We can understand accidents and assess risk by looking at the chain of events leading to the loss.

Some of the most important assumptions in safety lie in our models of how the world works. Models are important because they provide a means for understanding phenomena like accidents or potentially hazardous system behavior and for recording that understanding in a way that can be communicated to others.

A particular type of model, an *accident causality model* (or *accident model* for short) underlies all efforts to engineer for safety. Our accident models provide the foundation for (1) investigating and analyzing the cause of accidents, (2) designing to prevent future losses, and (3) assessing the risk associated with using the systems and products we create. Accident models explain why accidents occur, and they determine the approaches we take to prevent them. While you might not be consciously aware you are using a model when engaged in these activities, some (perhaps subconscious) model of the phenomenon is always part of the process.

All models are abstractions; they simplify the thing being modeled by abstracting away what are assumed to be irrelevant details and focusing on the features of the phenomenon that are judged to be the most relevant. Selecting some factors as relevant and others as irrelevant is, in most cases, arbitrary and entirely the choice of the modeler. That choice, however, is critical in determining the usefulness and accuracy of the model in predicting future events.

An underlying assumption of all accident models is that there are common patterns in accidents and that they are not simply random events. Accident models impose patterns on accidents and influence the factors considered in any safety analysis. Because the accident model influences what cause(s) is ascribed to an accident, the countermeasures taken to prevent future accidents, and the evaluation of the risk in operating a system, the power and features of the accident model used will greatly affect our ability to identify and control hazards and thus prevent accidents.

The earliest formal accident models came from *industrial safety* (sometimes called *occupational safety*) and reflect the factors inherent in protecting workers from injury or illness. Later, these same models or variants of them were applied to the engineering and operation of complex technical and social systems. At the beginning, the focus in industrial accident prevention was on unsafe conditions, such as open blades and unprotected belts. While this emphasis on preventing unsafe conditions was very successful in reducing workplace injuries, the decrease naturally started to slow down as the most obvious hazards were eliminated. The emphasis then shifted to unsafe acts: Accidents began to be regarded as someone's fault rather than as an event that could have been prevented by some change in the plant or product.

Heinrich's Domino Model, published in 1931, was one of the first published general accident models and was very influential in shifting the emphasis in safety to human error. Heinrich compared the general sequence of accidents to five dominoes, standing on end in a line (Figure 2.3). When the first domino falls, it automatically knocks down its neighbor and so on until the injury occurs. In any accident sequence, according to this model, ancestry or social environment leads to a fault of a person, which is the proximate reason for an unsafe act or condition (mechanical or physical), which results in an accident, which leads to an injury. In 1976, Bird and Loftus extended the basic Domino Model to include management decisions as a factor in accidents: in 1976:

1. Lack of control by management, permitting

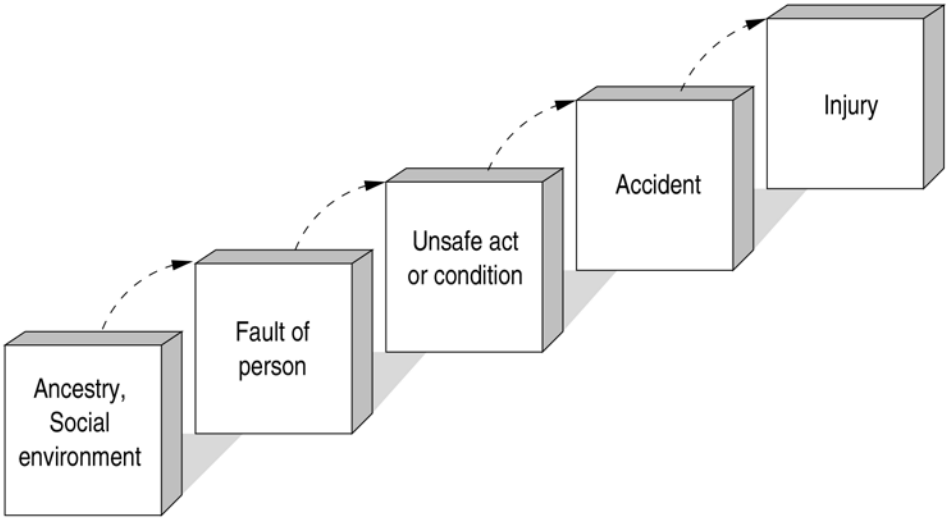


Figure 2.3: Heinrich's Domino Model of Accidents.

2. Basic causes (personal and job factors) that lead to
3. Immediate causes (substandard practices/conditions/errors) which are the proximate cause of
4. An accident or incident, which results in
5. A loss.

In the same year, Adams suggested a different management-augmented model that included:

1. Management structure (objectives, organization, and operations)
2. Operational errors (management or supervisory behavior)
3. Tactical errors (caused by employee behavior and work conditions)
4. Accident or incident
5. Injury or damage to persons or property.

Reason reinvented the Domino Model 20 years later in what he called the Swiss Cheese model, but with layers of Swiss cheese substituted for dominos and the layers or dominos labeled as layers of defense[3] that have failed [171, 172].

The basic Domino Model is inadequate for complex systems and other models were developed (see *Safeware* [114], Chapter 10), but the assumption that there is a single or *root cause* of an accident unfortunately persists as does the idea of dominos (or layers of Swiss cheese) and chains of failures, each directly causing or leading to the next one in the chain. It also lives on in the emphasis on human error in identifying accident causes.

The most common accident models today explain accidents in terms of multiple events sequenced as a forward chain over time. The events included almost always involve some type of "failure" event or human error, or they are energy related (for example, an explosion). The chains may be branching (as in fault trees) or there may be multiple chains synchronized by time or common

[3] Designing layers of defense is a common safety design approach used primarily in the process industry, particularly for nuclear power. Different design approaches are commonly used in other industries.

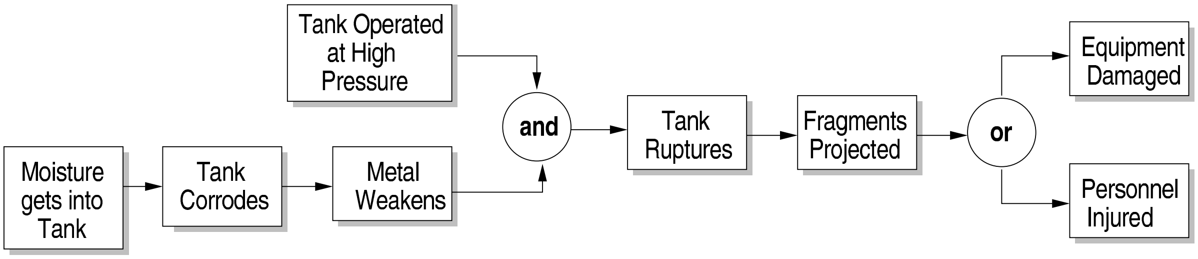


Figure 2.4: A model of the chain of events leading to the rupture of a pressurized tank (adapted from Hammer [78]). Moisture leads to corrosion, which causes weakened metal, which together with high operating pressures causes the tank to rupture, resulting in fragments being projected, and finally leading to personnel injury and/or equipment failure.

events. Lots of notations have been developed to represent the events in a graphical form, but the underlying model is the same. Figure 2.4 shows an example for the rupture of a pressurized tank.

The use of event-chain models of causation has important implications for the way engineers design for safety. If an accident is caused by a chain of events, then the most obvious preventive measure is to break the chain before the loss occurs. Because the most common events considered in these models are component failures, preventive measures tend to be focused on preventing failure events—increasing component integrity or introducing redundancy to reduce the likelihood of the event occurring. If corrosion can be prevented in the tank rupture accident, for example, then the tank rupture is averted.

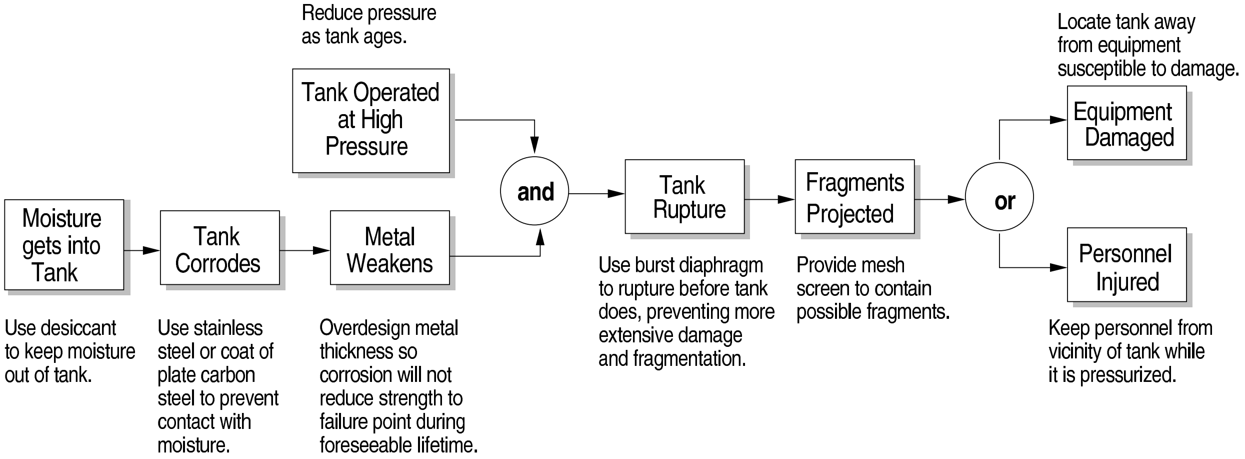

Reduce pressure as tank ages.
Tank Operated at High Pressure
Locate tank away from equipment susceptible to damage.
Equipment Damaged
Moisture gets into Tank
Tank Corrodes
Metal Weakens
and
Tank Rupture
Fragments Projected
or
Personnel Injured
Use desiccant to keep moisture out of tank.
Use stainless steel or coat of plate carbon steel to prevent contact with moisture.
Overdesign metal thickness so corrosion will not reduce strength to failure point during foreseeable lifetime.
Use burst diaphragm to rupture before tank does, preventing more extensive damage and fragmentation.
Provide mesh screen to contain possible fragments.
Keep personnel from vicinity of tank while it is pressurized.

Figure 2.5: The pressurized tank rupture event chain along with measures that could be taken to “break” the chain by preventing individual events in it.

Figure 2.5 is annotated with mitigation measures designed to break the chain. These mitigation measures are examples of the most common design techniques based on event-chain models of accidents, such as barriers (for example, preventing the contact of moisture with the metal used in the tank by coating it with plate carbon steel or providing mesh screens to contain fragments), interlocks (using a burst diaphragm), overdesign (increasing the metal thickness), and operational procedures (reducing the amount of pressure as the tank ages).

For this simple example involving only physical failures, designing to prevent such failures works well. But even this simple example omits any consideration of factors indirectly related to the events in the chain. An example of a possible indirect or systemic factor is competitive or financial pressures to increase efficiency that could lead to not following the plan to reduce the operating pressure as the tank ages. A second factor might be changes over time to the plant design that require workers to spend time near the tank while it is pressurized.

Formal and informal notations for representing the event chain may contain only the events or they may also contain the conditions that led to the events. Events create conditions that, along with existing conditions, lead to events that create new conditions and so on (Figure 2.6). The *tank corrodes* event leads to *corrosion exists in tank* condition which leads to *metal weakens* event which leads to weakened metal condition and so forth.

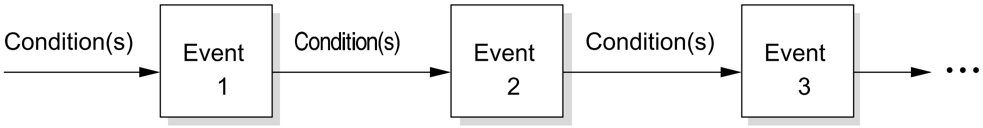


Figure 2.6: Conditions cause events which lead to new conditions which cause further events ...

The difference between events and conditions is that events are limited in time while conditions persist until some event occurs that results in new or changed conditions. For example, the three conditions that must exist before a flammable mixture will explode (the event) are the flammable gases or vapors themselves, air, and a source of ignition. Any one or two of these may exist for a period of time before the other(s) occurs and leads to the explosion. An event (the explosion) creates new conditions, such as uncontrolled energy or toxic chemicals in the air.

Causality models based on event chains (or dominos or layers of Swiss cheese) are simple and therefore appealing. But they are too simple and do not include what is needed to understand why accidents occur and how to prevent them. Some important limitations include requiring direct causality relationships, subjectivity in selecting the events to include, subjectivity in identifying chaining conditions, and exclusion of systemic factors.

2.2.1 Direct Causality

The causal relationships between the events in event chain models (or between dominoes or Swiss cheese slices) are required to be direct and linear, representing the notion that the preceding event must have occurred and the linking conditions must have been present for the subsequent event to occur: if event A had not occurred then the following event B would not have occurred. As such, event chain models encourage limited notions of linear causality, and it is difficult or impossible to incorporate nonlinear relationships. Consider the statement "Smoking causes lung cancer." Such a statement would not be allowed in the event-chain model of causality because there is no direct relationship between the two. Many smokers do not get lung cancer and some people who get lung cancer are not smokers. It is widely accepted, however, that there is some relationship between the two although it may be quite complex and nonlinear.

In addition to limitations in the types of causality considered, the causal factors identified using event-chain models depend on the events that are considered and on the selection of the conditions that link the events. Other than the physical events immediately preceding or directly involved in the loss, however, the choice of events to include is subjective and the conditions selected to explain the events is even more so. Each of these two limitations is considered in turn.

2.2.2 Subjectivity in Selecting Events

The selection of events to include in an event chain is dependent on the stopping rule used to determine how far back the sequence of explanatory events goes. Although the first event in the chain is often labeled the *initiating event* or *root cause*, the selection of an initiating event is arbitrary and previous events and conditions could always be added.

Sometimes the initiating event is selected (the backward chaining stops) because it represents a type of event that is familiar and thus acceptable as an explanation for the accident or it is a deviation from a standard [165]. In other cases, the initiating event or root cause is chosen because it is the first event in the backward chain for which it is felt that something can be done for correction.[4]

The backward chaining may also stop because the causal path disappears due to lack of information. Rasmussen suggests that a practical explanation for why actions by operators actively involved in the dynamic flow of events are so often identified as the cause of an accident is the difficulty in continuing the backtracking "through" a human [165].

A final reason why a "root cause" may be selected is that it is politically acceptable as the identified cause. Other events or explanations may be excluded or not examined in depth because they raise issues that are embarrassing to the organization or its contractors or are politically unacceptable.

The accident report on a friendly fire shootdown of a U.S. Army helicopter over the Iraqi no-fly zone in 1994, for example, describes the chain of events leading to the shootdown. Included in these events is the fact that the helicopter pilots did not change to the radio frequency required in the no-fly zone when they entered it (they stayed on the enroute frequency). Stopping at this event in the chain (which the official report does), it appears that the helicopter pilots were partially at fault for the loss by not following radio procedures. An independent account of the accident [158], however, notes that the U.S. commander of the operation had made an exception about the radio frequency to be used by the helicopters in order to mitigate a different safety concern (see Chapter 5), and therefore the pilots were simply following orders when they did not switch to the "required" frequency. The command to the helicopter pilots not to follow official radio procedures is not included in the chain of events provided in the official government accident report, but it suggests a very different understanding of the role of the helicopter pilots in the loss.

In addition to a *root* cause or causes, some events or conditions may be identified as *proximate* or *direct* causes while others are labeled as *contributory*. There is no more basis for this distinction than the selection of a root cause.

Making such distinctions between causes or limiting the factors considered can be a hindrance in learning from and preventing future accidents. Consider the following aircraft examples.

In the crash of an American Airlines DC-10 at Chicago's O'Hare Airport in 1979, the U.S. National Transportation Safety Board (NTSB) blamed only a "maintenance-induced crack," and not also a design error that allowed the slats to retract if the wing was punctured. Because of this

[4] As an example, a NASA Procedures and Guidelines document (NPG 8621 Draft 1) defines a root cause as: "Along a chain of events leading to an mishap, the first causal action or failure to act that could have been controlled systematically either by policy/practice/procedure or individual adherence to policy/practice/procedure."

omission, McDonnell Douglas was not required to change the design, leading to future accidents related to the same design error [154].

Similar omissions of causal factors in aircraft accidents have occurred more recently. One example is the crash of a China Airlines A300 on April 26, 1994, while approaching the Nagoya, Japan, airport. One of the factors involved in the accident was the design of the flight control computer software. Previous incidents with the same type of aircraft had led to a Service Bulletin being issued for a modification of the two flight control computers to fix the problem. But because the computer problem had not been labeled a "cause" of the previous incidents (for perhaps at least partially political reasons), the modification was labeled *recommended* rather than *mandatory.* China Airlines concluded, as a result, that the implementation of the changes to the computers was not urgent, and decided to delay modification until the next time the flight computers on the plane needed repair [4]. Because of that delay, 264 passengers and crew died.

In another DC-10 saga, explosive decompression played a critical role in a near miss over Windsor, Ontario. An American Airlines DC-10 lost part of its passenger floor, and thus all of the control cables that ran through it, when a cargo door opened in flight in June 1972. Thanks to the extraordinary skill and poise of the pilot, Bryce McCormick, the plane landed safely. In a remarkable coincidence, McCormick had trained himself to fly the plane using only the engines because he had been concerned about a decompression-caused collapse of the floor. After this close call, McCormick recommended that every DC-10 pilot be informed of the consequences of explosive decompression and trained in the flying techniques that he and his crew had used to save their passengers and aircraft. FAA investigators, the National Transportation Safety Board, and engineers at a subcontractor to McDonnell Douglas that designed the fuselage of the plane, all recommended changes in the design of the aircraft. Instead, McDonnell Douglas attributed the Windsor incident totally to human error on the part of the baggage handler responsible for closing the cargo compartment door (a convenient event in the event chain) and not to any error on the part of their designers or engineers and decided all they had to do was to come up with a fix that would prevent baggage handlers from forcing the door.

One of the discoveries after the Windsor incident was that the door could be improperly closed but the external signs, such as the position of the external handle, made it appear to be closed properly. In addition, this incident proved that the cockpit warning system could fail, and the crew would then not know that they were taking off without a properly closed door:

> The aviation industry does not normally receive such manifest warnings of basic design flaws in an aircraft without cost to human life. Windsor deserved to be celebrated as an exceptional case when every life was saved through a combination of crew skill and the sheer luck that the plane was so lightly loaded. If there had been more passengers and thus more weight, damage to the control cables would undoubtedly have been more severe, and it is highly questionable if any amount of skill could have saved the plane [60].

Almost two years later, in March 1974, a fully loaded Turkish Airlines DC-10 crashed near Paris resulting in 346 deaths—one of the worst accidents in aviation history. Once again, the cargo door had opened in flight, causing the cabin floor to collapse, severing the flight control cables. Immediately after the accident, Sanford McDonnell stated the official McDonnell-Douglas position that once again placed the blame on the baggage handler and the ground crew. This time, however, the FAA finally ordered modifications to all DC-10s that eliminated the hazard. In addition, an

FAA regulation issued in July 1975 required all wide-bodied jets to be able to tolerate a hole in the fuselage of twenty square feet. By labeling the root cause in the event chain as baggage handler error and attempting only to eliminate that event or link in the chain rather than the basic engineering design flaws, fixes that could have prevented the Paris crash were not made.

Until we do a better job of identifying causal factors in accidents, we will continue to have unnecessary repetition of incidents and accidents.

2.2.3 Subjectivity in Selecting the Chaining Conditions

In addition to subjectivity in selecting the events and the root cause event, the links between the events that are chosen to explain them are subjective and subject to bias. Leplat notes that the links are justified by knowledge or rules of different types, including physical and organizational knowledge. The same event can give rise to different types of links according to the mental representations the analyst has of the production of this event. When several types of rules are possible, the analyst will apply those that agree with his or her mental model of the situation [110].

Consider, for example, the loss of an American Airlines B-757 near Cali, Colombia, in 1995 [2]. Two significant events in this loss were

(1) *Pilot asks for clearance to take the* ROZO *approach*

followed later by

(2) *Pilot types R into the FMS.*[5]

In fact, the pilot should have typed the four letters ROZO instead of *R*—the latter was the symbol for a different radio beacon (called ROMEO) near Bogota. As a result, the aircraft incorrectly turned toward mountainous terrain. While these events are noncontroversial, the link between the two events could be explained by any of the following:

- *Pilot Error*: In the rush to start the descent, the pilot executed a change of course without verifying its effect on the flight path.
- *Crew Procedure Error*: In the rush to start the descent, the captain entered the name of the waypoint without normal verification from the other pilot.
- *Approach Chart and FMS Inconsistencies*: The identifier used to identify ROZO on the approach chart (*R*) did not match the identifier used to call up ROZO in the FMS.
- *FMS Design Deficiency*: The FMS did not provide the pilot with feedback that choosing the first identifier listed on the display was not the closest beacon having that identifier.
- *American Airlines Training Deficiency*: The pilots flying into South America were not warned about duplicate beacon identifiers nor adequately trained on the logic and priorities used in the FMS on the aircraft.

[5] An FMS is an automated Flight Management System, which assists the pilots in various ways. In this case, it was being used to provide navigation information.

- *Manufacturer Deficiency*: Jeppesen-Sanderson did not inform airlines operating FMS-equipped aircraft of the differences between navigation information provided by Jeppesen-Sanderson Flight Management System navigation databases and Jeppesen-Sanderson approach charts or the logic and priorities employed in the display of electronic FMS navigation information.

- *International Standards Deficiency*: No single worldwide standard provides unified criteria for the providers of electronic navigation databases used in Flight Management Systems.

The selection of the linking condition (or events) will greatly influence the cause ascribed to the accident yet in the example all are plausible and each could serve as an explanation of the event sequence. The choice may reflect more on the person or group making the selection than on the accident itself. In fact, understanding this accident and learning enough from it to prevent future accidents requires identifying *all* of these factors to explain the incorrect input: The accident model used should encourage and guide a comprehensive analysis at multiple technical and social system levels.

2.2.4 Discounting Systemic Factors

The problem with event chain models is not simply that the selection of the events to include and the labeling of some of them as causes are arbitrary or that the selection of which conditions to include is also arbitrary and usually incomplete. Even more important is that viewing accidents as chains of events and conditions may limit understanding and learning from the loss and omit causal factors that cannot be included in an event chain.

Event chains developed to explain an accident usually concentrate on the proximate events immediately preceding the loss. But the foundation for an accident is often laid years before. One event simply triggers the loss, but if that event had not happened, another one would have led to a loss. The Bhopal disaster provides a good example.

The release of methyl isocyanate (MIC) from the Union Carbide chemical plant in Bhopal, India, in December 1984 has been called the worst industrial accident in history: Conservative estimates point to 2,000 fatalities, 10,000 permanent disabilities (including blindness), and 200,000 injuries [37]. The Indian government blamed the accident on human error—the improper cleaning of a pipe at the plant. A relatively new worker was assigned to wash out some pipes and filters, which were clogged. MIC produces large amounts of heat when in contact with water, and the worker properly closed the valves to isolate the MIC tanks from the pipes and filters being washed. Nobody, however, inserted a required safety disk (called a *slip blind*) to back up the valves in case they leaked [12].

A chain of events describing the accident mechanism for Bhopal might include:

E1 Worker washes pipes without inserting a slip blind.

E2 Water leaks into MIC tank.

E3 Explosion occurs.

E4 Relief valve opens.

E5 MIC vented into air.

E6 Wind carries MIC into populated area around plant.

Both Union Carbide and the Indian government blamed the worker washing the pipes for the accident.[6] A different operator error might be identified as the root cause (initiating event) if the chain is followed back farther. The worker who had been assigned the task of washing the pipes reportedly knew that the valves leaked, but he did not check to see whether the pipe was properly isolated because, he said, it was not his job to do so. Inserting the safety disks was the job of the maintenance department, but the maintenance sheet contained no instruction to insert this disk. The pipe-washing operation should have been supervised by the second shift supervisor, but that position had been eliminated in a cost-cutting effort. So the root cause might instead have been assigned to the person responsible for inserting the slip blind or to the second shift supervisor.

But the selection of a stopping point and the specific operator action to label as the root cause—and operator actions are almost always selected as root causes—is not the real problem here. The problem is the oversimplification implicit in using a chain of events to understand why this accident occurred: Given the design and operating conditions of the plant, an accident was waiting to happen:

> However [water] got in, it would not have caused the severe explosion had the refrigeration unit not been disconnected and drained of freon, or had the gauges been properly working and monitored, or had various steps been taken at the first smell of MIC instead of being put off until after the tea break, or had the scrubber been in service, or had the water sprays been designed to go high enough to douse the emissions, or had the flare tower been working and been of sufficient capacity to handle a large excursion [155, p.349].

It is not uncommon for a company to turn off passive safety devices, such as refrigeration units, to save money. The operating manual specified that the refrigeration unit *must* be operating whenever MIC was in the system: The chemical has to be maintained at a temperature no higher than 5° Celsius to avoid uncontrolled reactions. A high temperature alarm was to sound if the MIC reached 11°. The refrigeration unit was turned off, however, to save money and the MIC was usually stored at nearly 20°. The plant management adjusted the threshold of the alarm, accordingly, from 11° to 20° and logging of tank temperatures was halted, thus eliminating the possibility of an early warning of rising temperatures.

Gauges at plants are frequently out of service [22]. At the Bhopal facility, there were few alarms or interlock devices in critical locations that might have warned operators of abnormal conditions—a system design deficiency.

Other protection devices at the plant had inadequate design thresholds. The vent scrubber, had it worked, was designed to neutralize only small quantities of gas at fairly low pressures and temperatures: The pressure of the escaping gas during the accident exceeded the scrubber's design by nearly two and a half times, and the temperature of the escaping gas was at least 80° Celsius more than the scrubber could handle. Similarly, the flare tower (which was supposed to burn off released vapor) was totally inadequate to deal with the estimated 40 tons of MIC that escaped during the accident. In addition, the MIC was vented from the vent stack 108 feet above the ground, well above the height of the water curtain intended to knock down the gas: The water

[6] Union Carbide lawyers argued that the introduction of water into the MIC tank was an act of sabotage rather than a maintenance worker's mistake. While this differing interpretation of the initiating event has important implications with respect to legal liability, it makes no difference in the argument presented here regarding the limitations of event-chain models of accidents or even, as will be seen, understanding why this accident occurred.

curtain reached only 40 to 50 feet above the ground. The water jets could reach as high as 115 feet, but only if operated individually.

Leaks were routine occurrences and the reasons for them were seldom investigated: Problems were either fixed without further examination or were ignored. A safety audit two years earlier by a team from Union Carbide had noted many safety problems at the plant, including several involved in the accident, such as filter-cleaning operations without using slip blinds, leaking valves, the possibility of contaminating the tank with material from the vent gas scrubber, and bad pressure gauges. The safety auditors had recommended increasing the capability of the water curtain and had pointed out that the alarm at the flare tower from which the MIC leaked was nonoperational and thus any leak could go unnoticed for a long time. None of the recommended changes were made [22]. There is debate about whether the audit information was fully shared with the Union Carbide India subsidiary and about who was responsible for making sure changes were made. In any event, there was no follow-up to make sure that the problems identified in the audit had been corrected.

A year before the accident, the chemical engineer managing the MIC plant resigned because he disapproved of falling safety standards and still no changes were made. He was replaced by an electrical engineer.

Measures for dealing with a chemical release once it occurred were no better. Alarms at the plant sounded so often (the siren went off 20 to 30 times a week for various purposes) that an actual alert could not be distinguished from routine events or practice alerts. Ironically, the warning siren was not turned on until two hours after the MIC leak was detected (and after almost all the injuries had occurred) and then was turned off after only five minutes—which was company policy [12]. Moreover, the numerous practice alerts did not seem to be effective in preparing for an emergency: When the danger during the release became known, many employees ran from the contaminated areas of the plant, totally ignoring the buses that were sitting idle ready to evacuate workers and nearby residents. Plant workers had only a bare minimum of emergency equipment—a shortage of oxygen masks, for example, was discovered after the accident started—and they had almost no knowledge or training about how to handle nonroutine events.

The police were not notified when the chemical release began; in fact, when called by police and reporters, plant spokesmen first denied the accident and then claimed that MIC was not dangerous. Nor was the surrounding community warned of the dangers, before or during the release, or informed of the simple precautions that could have saved them from lethal exposure, such as putting a wet cloth over their face and closing their eyes. If the community had been alerted and provided with this simple information, many (if not most) lives would have been saved and injuries prevented [105].

Some of the reasons why the poor conditions in the plant were allowed to persist are financial. Demand for MIC had dropped sharply after 1981, leading to reductions in production and pressure on the company to cut costs. The plant was operating at less than half capacity when the accident occurred. Union Carbide put pressure on the Indian management to reduce losses, but gave no specific details on how to achieve the reductions. In response, the maintenance and operating personnel were cut in half. Maintenance procedures were severely cut back and the shift relieving system was suspended—if no replacement showed up at the end of the shift, the following shift went unmanned. The person responsible for inserting the slip blind in the pipe had not showed up for his shift. Top management justified the cuts as merely reducing avoidable and wasteful expenditures without affecting overall safety.

As the plant lost money, many of the skilled workers left for more secure jobs. They either were not replaced or were replaced by unskilled workers. When the plant was first built, operators and technicians had the equivalent of two years of college education in chemistry or chemical engineering. In addition, Union Carbide provided them with six months training. When the plant began to lose money, educational standards and staffing levels were reportedly reduced. In the past, UC flew plant personnel to West Virginia for intensive training and had teams of U.S. engineers make regular on-site safety inspections. But by 1982, financial pressures led UC to give up direct supervision of safety at the plant, even though it retained general financial and technical control. No American advisors were resident at Bhopal after 1982.

Management and labor problems followed the financial losses. Morale at the plant was low. "There was widespread belief among employees that the management had taken drastic and imprudent measures to cut costs and that attention to details that ensure safe operation were absent" [126].

These are only a few of the factors involved in this catastrophe, which also include other technical and human errors within the plant, design errors, management negligence, regulatory deficiencies on the part of the U.S. and Indian governments, and general agricultural and technology transfer policies related to the reason they were making such a dangerous chemical in India in the first place. Any one of these perspectives or "causes" is inadequate by itself to understand the accident and to prevent future ones. In particular, identifying only operator error or sabotage as the root cause of the accident ignores most of the opportunities for the prevention of similar accidents in the future. Many of the systemic causal factors are only indirectly related to the proximate events and conditions preceding the loss.

When all the factors, including indirect and systemic ones, are considered, it becomes clear that the maintenance worker was, in fact, only a minor and somewhat irrelevant player in the loss. Instead, degradation in the safety margin occurred over time and without any particular single decision to do so but simply as a series of decisions that moved the plant slowly toward a situation where any slight error would lead to a major accident. Given the overall state of the Bhopal Union Carbide plant and its operation, if the action of inserting the slip disk had not been left out of the pipe washing operation that December day in 1984, something else would have triggered an accident. In fact, a similar leak had occurred the year before, but did not have the same catastrophic consequences and the true root causes of that incident were neither identified nor fixed.

To label one event (such as a maintenance worker leaving out the slip disk) or even several events as the root cause or the start of an event chain leading to the Bhopal accident is misleading at best. Rasmussen writes:

> The stage for an accidental course of events very likely is prepared through time by the normal efforts of many actors in their respective daily work context, responding to the standing request to be more productive and less costly. Ultimately, a quite normal variation in somebody's behavior can then release an accident. Had this 'root cause' been avoided by some additional safety measure, the accident would very likely be released by another cause at another point in time. In other words, an explanation of the accident in terms of events, acts, and errors is not very useful for design of improved systems [166].

In general, event-based models are poor at representing systemic accident factors such as structural deficiencies in the organization, management decision making, and flaws in the safety culture of the company or industry. An accident model should encourage a broad view of accident mechanisms that expands the investigation beyond the proximate events: A narrow focus on technological components and pure engineering activities or a similar narrow focus on operator errors may lead to ignoring some of the most important factors in terms of preventing future accidents. The accident model used to explain why the accident occurred should not only encourage the inclusion of all the causal factors but should provide guidance in identifying these factors.

2.2.5 Including Systems Factors in Accident Models

Large-scale engineered systems are more than just a collection of technological artifacts: They are a reflection of the structure, management, procedures, and culture of the engineering organization that created them. They are usually also a reflection of the society in which they were created. Ralph Miles Jr., in describing the basic concepts of systems theory, notes that:

> Underlying every technology is at least one basic science, although the technology may be well developed long before the science emerges. Overlying every technical or civil system is a social system that provides purpose, goals, and decision criteria. [136, p.1]

Effectively preventing accidents in complex systems requires using accident models that include that social system as well as the technology and its underlying science. Without understanding the purpose, goals, and decision criteria used to construct and operate systems, it is not possible to completely understand and most effectively prevent accidents.

Awareness of the importance of social and organizational aspects of safety goes back to the early days of System Safety[7]. In 1968, Jerome Lederer, then the director of the NASA Manned Flight Safety Program for Apollo, wrote:

> System safety covers the total spectrum of risk management. It goes *beyond the hardware* and associated procedures of system safety engineering. It involves: attitudes and motivation of designers and production people, employee/management rapport, the relation of industrial associations among themselves and with government, human factors in supervision and quality control, documentation on the interfaces of industrial and public safety with design and operations, the interest and attitudes of top management, the effects of the legal system on accident investigations and exchange of information, the certification of critical workers, political considerations, resources, public sentiment and many other non-technical but vital influences on the attainment of an acceptable level of risk control. These non-technical aspects of system safety cannot be ignored [108].

Too often, however, these non-technical aspects are ignored.

At least three types of factors need to be considered in accident causation. The first is the proximate event chain, which for the *Herald of Free Enterprise* includes the assistant bosun not closing the doors and the return of the first officer to the wheelhouse prematurely. Note that there was a redundant design here, with the first officer checking the work of the assistant bosun, but it did not prevent the accident as is often the case with redundancy [114, 154].

[7] When this term is capitalized in this book, it denotes the specific form of safety engineering developed originally by the Defense Department and its contractors for the early ICBM systems and defined by MIL-STD-882. System safety (uncapitalized) or safety engineering denotes all the approaches to engineering for safety.

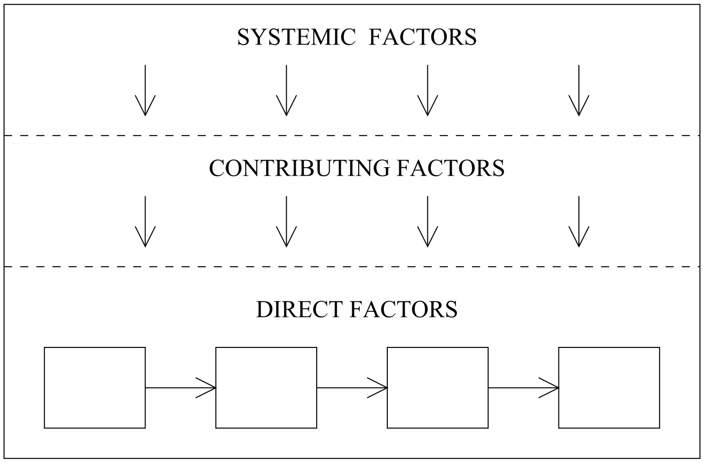


Figure 2.7: Johnson's Three Level Model of Accidents

The second type of information includes the conditions that allowed the events to occur, i.e., the high spring tides, the inadequate design of the ferry loading ramp for this harbor, and the desire of the first officer to stay on schedule (thus leaving the car deck before the doors were closed). All of these conditions can be directly mapped to the events.

The third set of causal factors is only indirectly related to the events and conditions, but these indirect factors are critical in fully understanding why the accident occurred and thus how to prevent future accidents. In this case, the systemic factors include the owner of the ferry (Townsend Thoresen) needing ships that were designed to permit fast loading and unloading and quick acceleration in order to remain competitive in the ferry business, and pressure by company management on the captain and first officer to strictly adhere to schedules, also related to competitive factors.

Several attempts have been made to graft systemic factors onto event models, but all have important limitations. The most common approach has been to add hierarchical levels above the event chain. In the seventies, Johnson proposed a model and sequencing method that described accidents as chains of direct events and causal factors arising from contributory factors, which in turn arise from systemic factors (Figure 2.7) [92].

Johnson also tried to put management factors into fault trees (a technique called MORT or Management Oversight Risk Tree), but ended up simply providing a general checklist for auditing management practices. While such a checklist can be very useful, it presupposes that every error can be predefined and put into a checklist form. The checklist is comprised of a set of questions that should be asked during an accident investigation. Examples of the questions from a DOE MORT User's Manual are: Was there sufficient training to update and improve needed supervisory skills? Did the supervisors have their own technical staff or access to such individuals? Was there technical support of the right discipline(s) sufficient for the needs of supervisory programs and review functions? Were there established methods for measuring performance that permitted the effectiveness of supervisory programs to be evaluated? Was a maintenance plan provided before startup? Was all relevant information provided to planners and managers? Was it used? Was concern for safety displayed by vigorous, visible personal action by top executives? And so forth.

Johnson originally provided hundreds of such questions, and additions have been made to his checklist since Johnson created it in the 1970s so it is now even larger. The use of the MORT checklist is feasible because the items are so general, but that same generality also limits its usefulness. Something more effective than checklists is needed.

The most sophisticated of the hierarchical add-ons to event chains is Rasmussen and Svedung's model of the sociotechnical system involved in risk management [166]. As shown in Figure 2.8, at the social and organizational levels they use a hierarchical control structure, with levels for government, regulators and associations, company, management, and staff. At all levels they map information flow. The model concentrates on operations; information from the system design and analysis process is treated as input to the operations process. At each level, they model the factors involved using event chains, with links to the event chains at the level below. Notice that they still assume there is a root cause and causal chain of events. A generalization of the Rasmussen and Svedung model, which overcomes these limitations, is presented in Chapter 4.

Once again, a new assumption is needed to make progress in learning how to design and operate safer systems:

New Assumption 2: Accidents are complex processes involving the entire sociotechnical system. Traditional event-chain models cannot describe this process adequately.

Most of the accident models underlying safety engineering today stem from the days when the types of systems we were building and the context in which they were built were much simpler. As noted in Chapter 1, new technology and social factors are making fundamental changes in the etiology of accidents, requiring changes in the explanatory mechanisms used to understand them and in the engineering techniques applied to prevent them.

Event-based models are limited in their ability to represent accidents as complex processes, particularly at representing systemic accident factors such as structural deficiencies in the organization, management deficiencies, and flaws in the safety culture of the company or industry. We need to understand how the whole system, including the organizational and social components, operating together, led to the loss. While some extensions to event-chain models have been proposed, all are unsatisfactory in important ways.

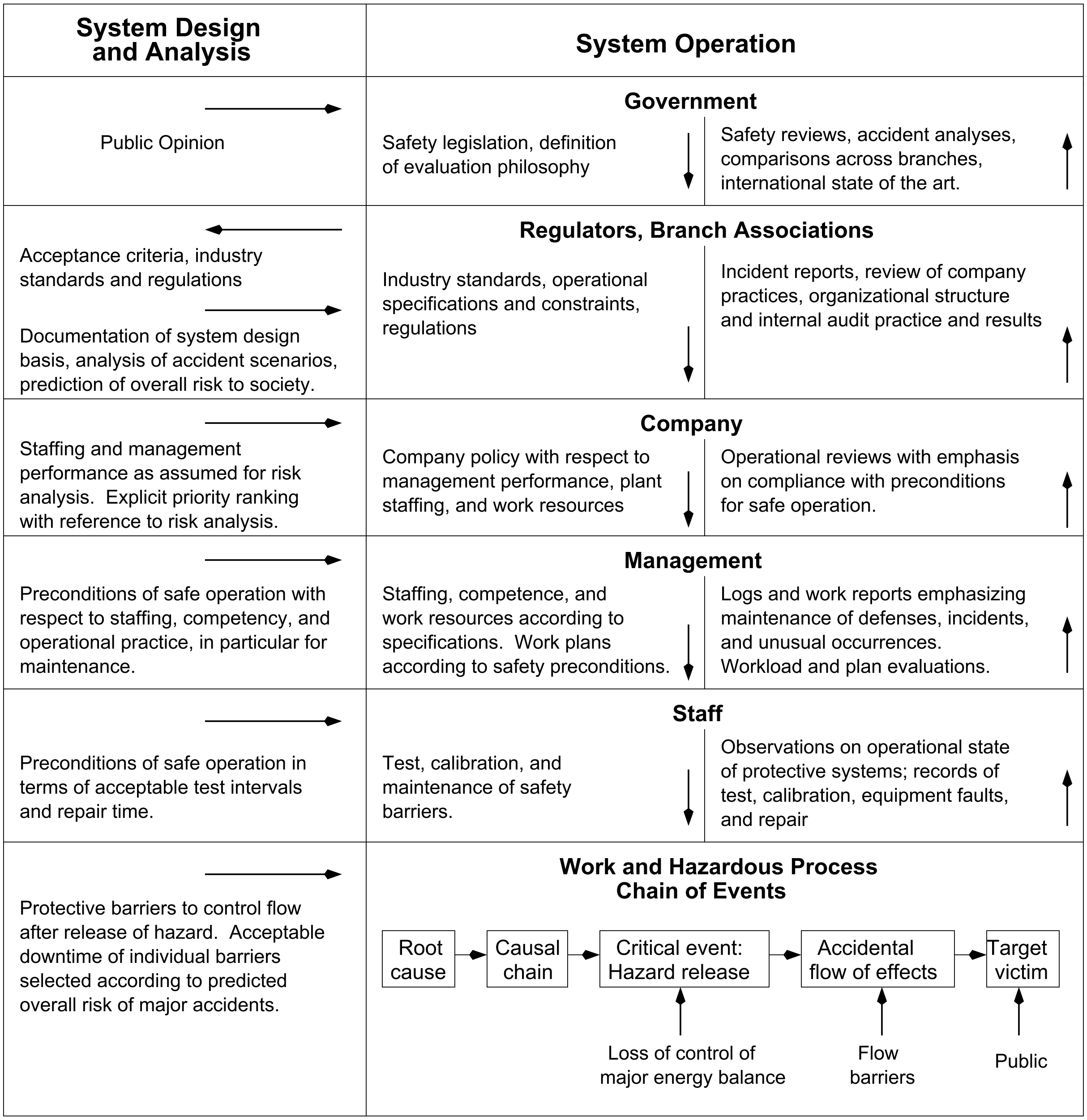
An accident model should encourage a broad view of accident mechanisms that expands the investigation beyond the proximate events: A narrow focus on operator actions, physical component failures, and technology may lead to ignoring some of the most important factors in terms of preventing future accidents. The whole concept of "root cause" needs to be reconsidered.

2.3 Limitations of Probabilistic Risk Assessment

Assumption 3: Probabilistic risk analysis based on event chains is the best way to assess and communicate safety and risk information.

The limitations of event-chain models are reflected in the current approaches to quantitative risk assessment, most of which use trees or other forms of event chains. Probabilities (or probability density functions) are assigned to the events in the chain and an overall likelihood of a loss is calculated.

In performing a probabilistic risk assessment (PRA), initiating events in the chain are usually assumed to be mutually exclusive. While this assumption simplifies the mathematics, it may not



System Design and Analysis	System Operation	
	Government	
Public Opinion	Safety legislation, definition of evaluation philosophy	Safety reviews, accident analyses, comparisons across branches, international state of the art.
	Regulators, Branch Associations	
Acceptance criteria, industry standards and regulations Documentation of system design basis, analysis of accident scenarios, prediction of overall risk to society.	Industry standards, operational specifications and constraints, regulations	Incident reports, review of company practices, organizational structure and internal audit practice and results
	Company	
Staffing and management performance as assumed for risk analysis. Explicit priority ranking with reference to risk analysis.	Company policy with respect to management performance, plant staffing, and work resources	Operational reviews with emphasis on compliance with preconditions for safe operation.
	Management	
Preconditions of safe operation with respect to staffing, competency, and operational practice, in particular for maintenance.	Staffing, competence, and work resources according to specifications. Work plans according to safety preconditions.	Logs and work reports emphasizing maintenance of defenses, incidents, and unusual occurrences. Workload and plan evaluations.
	Staff	
Preconditions of safe operation in terms of acceptable test intervals and repair time.	Test, calibration, and maintenance of safety barriers.	Observations on operational state of protective systems; records of test, calibration, equipment faults, and repair
	Work and Hazardous Process Chain of Events	
Protective barriers to control flow after release of hazard. Acceptable downtime of individual barriers selected according to predicted overall risk of major accidents.		

Root cause
Causal chain
Critical event: Hazard release
Accidental flow of effects
Target victim
Loss of control of major energy balance
Flow barriers
Public

Figure 2.8: The Rasmussen/Svedung Model of Risk Management

match reality. As an example, consider the following description of an accident chain for an offshore oil platform:

> An initiating event is an event that triggers an accident sequence—e.g., a wave that exceeds the jacket's capacity that, in turn, triggers a blowout that causes failures of the foundation. As initiating events, they are mutually exclusive; only one of them starts the accident sequence. A catastrophic platform failure can start by failure of the foundation, failure of the jacket, or failure of the deck. These initiating failures are also (by definition) mutually exclusive and constitute the basic events of the [Probabilistic Risk Assessment] model in its simplest form [151, p.121].

The selection of the failure of the foundation, jacket, or deck as the initiating event is arbitrary, as we have seen, and eliminates from consideration prior events leading to them such as manufacturing or construction problems. The failure of the foundation, for example, might be related to the use of inferior construction materials which, in turn, might be related to budget deficiencies or lack of government oversight.

In addition, there does not seem to be any reason for assuming that initiating failures are mutually exclusive and that only one starts the accident, except perhaps again to simplify the mathematics. In accidents, seemingly independent failures may have a common systemic cause (often not a failure) that results in coincident failures. For example, the same pressures to use inferior materials in the foundation may result in their use in the jacket and the deck, leading to a wave causing coincident, dependent failures in all three. Alternatively, the design of the foundation—a systemic factor rather than a failure event—may lead to pressures on the jacket and deck when stresses cause deformities in the foundation. Treating such events as independent may lead to unrealistic risk assessments.

In the Bhopal accident, the vent scrubber, flare tower, water spouts, refrigeration unit, and various monitoring instruments were all out of operation simultaneously. Assigning probabilities to all these seemingly unrelated events and assuming independence would lead one to believe that this accident was merely a matter of a once-in-a-lifetime coincidence. A probabilistic risk assessment based on an event chain model most likely would have treated these conditions as independent failures and then calculated their coincidence as being so remote as to be beyond consideration. Reason, in his popular Swiss Cheese Model of accident causation based on defense in depth, does the same, arguing that in general "the chances of such a trajectory of opportunity finding loopholes in all the defences [sic] at any one time is very small indeed" [171, p. 208]. As suggested earlier, a closer look at Bhopal and, indeed, most accidents paints a quite different picture and shows these were not random failure events but were related to engineering and management decisions stemming from common systemic factors.

Most accidents in well-designed systems involve two or more low-probability events occurring in the worst possible combination. When people attempt to predict system risk, they explicitly or implicitly multiply events with low probability—assuming independence—and come out with impossibly small numbers, when, in fact, the events are dependent. This dependence may be related to common systemic factors that do not appear in an event chain. Machol calls this phenomenon the *Titanic coincidence* [130].[8].

[8] Watt defined a related phenomenon he called the *Titanic effect* to explain the fact that major accidents are often preceded by a belief that they cannot happen. The *Titanic effect* says that the magnitude of disasters decreases to the extent that people believe that disasters are possible and plan to prevent them or to minimize their effects [204]

A number of "coincidences" contributed to the *Titanic* accident and the subsequent loss of life. For example, the captain was going far too fast for existing conditions, a proper watch for icebergs was not kept, the ship was not carrying enough lifeboats, lifeboat drills were not held, the lifeboats were lowered properly but arrangements for manning them were insufficient, and the radio operator on a nearby ship was asleep and so did not hear the distress call. Many of these events or conditions may be considered independent but appear less so when we consider that overconfidence due to incorrect engineering analyses about the safety and unsinkability of the ship most likely contributed to the excessive speed, the lack of a proper watch, and the insufficient number of lifeboats and drills. That the collision occurred at night contributed to the iceberg not being easily seen, made abandoning ship more difficult than it would have been during the day, and was a factor in why the nearby ship's operator was asleep [134]. Assuming independence here leads to a large underestimate of the true risk.

Another problem in probabilistic risk assessment (PRA) is the emphasis on failure events—design errors are usually omitted and only come into the calculation indirectly through the probability of the failure event. Accidents involving dysfunctional interactions among non-failing (operational) components—that is, component interaction accidents—are usually not considered. Systemic factors also are not reflected. In the offshore oil platform example at the beginning of this section, the true probability density function for the failure of the deck might reflect a poor design for the conditions the deck must withstand (a human design error) or, as noted earlier, the use of inadequate construction materials due to lack of government oversight or project budget limitations.

When historical data are used to determine the failure probabilities used in the PRA, non-failure factors, such as design errors or unsafe management decisions may differ between the historic systems from which the data was derived and the system under consideration. It is possible (and obviously desirable) for each PRA to include a description of the conditions under which the probabilities were derived. If such a description is not included, it may not be possible to determine whether conditions in the platform being evaluated differ from those built previously that might significantly alter the risk. The introduction of a new design feature or of active control by a computer might greatly affect the probability of failure and the usefulness of data from previous experience then becomes highly questionable.

The most dangerous result of using PRA arises from considering only immediate physical failures. Latent design errors may be ignored and go uncorrected due to overconfidence in the risk assessment. An example, which is a common but dangerous practice judging from its implication in a surprising number of accidents, is wiring a valve to detect only that power has been applied to open or close it and not that the valve position has actually changed. In one case, an Air Force system included a relief valve to be opened by the operator to protect against overpressurization [3]. A second, backup relief valve was installed in case the primary valve failed. The operator needed to know that the first valve had not opened, however, in order to determine that the backup valve must be activated. One day, the operator issued a command to open the primary valve. The position indicator and open indicator lights both illuminated but the primary relief valve was *not* open. The operator, thinking the primary valve had opened, did not activate the backup valve and an explosion occurred.

A post-accident investigation discovered that the indicator light circuit was wired to indicate *presence of power* at the valve, but it did not indicate valve *position*. Thus, the indicator showed only that the activation button had been pushed, not that the valve had operated. An extensive probabilistic risk assessment of this design had correctly assumed a low probability of simultaneous failure for the two relief valves, but had ignored the possibility of a design error in the electrical wiring: The probability of that design error was not quantifiable. If it had been identified, of course, the proper solution would have been to eliminate the design error, not to assign a probability to it. The same type of design flaw was a factor in the Three Mile Island accident: An indicator misleadingly showed that a discharge valve had been ordered closed but not that it had actually closed. In fact, the valve was blocked in an open position.

In addition to these limitations of PRA for electromechanical systems, current methods for quantifying risk that are based on combining probabilities of individual component failures and mutually exclusive events are not appropriate for systems controlled by software and by humans making cognitively complex decisions, and there is no effective way to incorporate management and organizational factors, such as flaws in the safety culture, despite many well-intentioned efforts to do so. As a result, these critical factors in accidents are often omitted from risk assessment because analysts do not know how to obtain a "failure" probability, or alternatively, a number is pulled out of the air for convenience. If we knew enough to measure these types of design flaws, it would be better to fix them than to try to measure them.

Another possibility for future progress is usually not considered:

New Assumption 3: Risk and safety may be best understood and communicated in ways other than probabilistic risk analysis.

Understanding risk is important in decision making. Many people assume that risk information is most appropriately communicated in the form of a probability. Much has been written, however, about the difficulty people have in interpreting probabilities [96]. Even if people could use such values appropriately, the tools commonly used to compute these quantities, which are based on computing probabilities of failure events, have serious limitations. An accident model that is not based on failure events, such as the one introduced in this book, could provide an entirely new basis for understanding and evaluating safety and, more generally, risk.

2.4 The Role of Operators in Accidents

Assumption 4: Most accidents are caused by operator error. Rewarding safe behavior and punishing unsafe behavior will eliminate or reduce accidents significantly.

As we have seen, the definition of "caused by" is debatable. But the fact remains that if there are operators in the system, they are most likely to be blamed for an accident. This phenomenon is not new. In the nineteenth century, coupling accidents on railroads were one of the main causes of injury and death to railroad workers [78]. In the seven years between 1888 and 1894, 16,000 railroad workers were killed in coupling accidents and 170,000 were crippled. Managers claimed that such accidents were due only to worker error and negligence, and therefore nothing could be done aside from telling workers to be more careful. The government finally stepped in and required that automatic couplers be installed. As a result, fatalities dropped sharply. According to the June 1896 (three years after Congress acted on the problem) issue of *Scientific American*:

> Few battles in history show so ghastly a fatality. A large percentage of these deaths

> were caused by the use of imperfect equipment by the railroad companies; twenty years ago it was practically demonstrated that cars could be automatically coupled, and that it was no longer necessary for a railroad employee to imperil his life by stepping between two cars about to be connected. In response to appeals from all over, the U.S. Congress passed the Safety Appliance Act in March 1893. It has or will cost the railroads $50,000,000 to fully comply with the provisions of the law. Such progress has already been made that the death rate has dropped by 35 per cent.

2.4.1 Do Operators Cause Most Accidents?

The tendency to blame the operator is not simply a nineteenth century problem, but persists today. During and after World War II, the Air Force had serious problems with aircraft accidents: From 1952 to 1966, for example, 7715 aircraft were lost and 8547 people killed [78]. Most of these accidents were blamed on pilots. Some aerospace engineers in the 1950s did not believe the cause was so simple and argued that safety must be designed and built into aircraft just as are performance, stability, and structural integrity. Although a few seminars were conducted and papers written about this approach, the Air Force did not take it seriously until they began to develop intercontinental ballistic missiles: there were no pilots to blame for the frequent and devastating explosions of these liquid-propellant missiles. In having to confront factors other than pilot error, the Air Force began to treat safety as a system problem, and System Safety programs were developed to deal with them. Similar adjustments in attitude and practice may be forced in the future by the increasing use of unmanned autonomous aircraft and other automated systems.

It is still common to see statements that 70 percent to 80 percent of aircraft accidents are caused by pilot error or that 85 percent of work accidents are due to unsafe acts by workers rather than unsafe conditions. However, closer examination shows that the data may be biased and incomplete: the less that is known about an accident, the most likely it will be attributed to operator error [92]. Thorough investigation of serious accidents almost invariably finds other factors.

Part of the problem stems from the use of the chain-of-events model in accident investigation because it is difficult to find an *event* preceding and causal to the operator behavior, as mentioned earlier. If the problem is in the system design, there is no proximal event to explain the error, only a flawed decision during system design.

Even if a technical failure precedes the human action, the tendency is to put the blame on an inadequate response to the failure by an operator. Perrow claims that even in the best of industries, there is rampant attribution of accidents to operator error, to the neglect of errors by designers or managers [154]. He cites a U.S. Air Force study of aviation accidents demonstrating that the designation of human error (pilot error in this case) is a convenient classification for mishaps whose real cause is uncertain, complex, or embarrassing to the organization.

Beside the fact that operator actions represent a convenient stopping point in an event chain, other reasons for the operator error statistics include: (1) operator actions are generally reported only when they have a negative impact on safety and not when they are responsible for preventing accidents; (2) blame may be based on unrealistic expectations that operators can overcome every emergency; (3) operators may have to intervene at the limits of system behavior when the consequences of not succeeding are likely to be serious and often involve a situation the designer never anticipated and was not covered by the operator's training; and (4) hindsight often allows us to identify a better decision in retrospect, but detecting and correcting potential errors before they have been made obvious by an accident is far more difficult.[9]

2.4.2 Hindsight Bias

The psychological phenomenon called *hindsight bias* plays such an important role in attribution of causes to accidents that it is worth spending time on it. The report on the Clapham Junction railway accident in Britain concluded:

> There is almost no human action or decision that cannot be made to look flawed and less sensible in the misleading light of hindsight. It is essential that the critic should keep himself constantly aware of that fact [81, pg. 147].

After an accident, it is easy to see where people went wrong, what they should have done or not done, to judge people for missing a piece of information that turned out to be critical, and to see exactly the kind of harm that they should have foreseen or prevented [50]. Before the event, such insight is difficult and, perhaps, impossible.

Dekker [50] points out that hindsight allows us to:

- Oversimplify causality because we can start from the outcome and reason backwards to presumed or plausible "causes."

- Overestimate the likelihood of the outcome—and people's ability to foresee it—because we already know what the outcome is.

- Overrate the role of rule or procedure "violations." There is always a gap between written guidance and actual practice, but this gap almost never leads to trouble. It only takes on causal significance once we have a bad outcome to look at and reason about.

- Misjudge the prominence or relevance of data presented to people at the time.

- Match outcome with the actions that went before it. If the outcome was bad, then the actions leading up to it must have also been bad—missed opportunities, bad assessments, wrong decisions, and misperceptions.

Avoiding hindsight bias requires changing our emphasis in analyzing the role of humans in accidents from what they did wrong to why it made sense for them to act the way they did.

2.4.3 The Impact of System Design on Human Error

All human activity takes place within and is influenced by the environment, both physical and social, in which it takes place. It is, therefore, often very difficult to separate system design error from operator error: In highly automated systems, the operator is often at the mercy of the system design and operational procedures. One of the major mistakes made by the operators at Three Mile Island was following the procedures provided to them by the utility. The instrumentation design also did not provide the information they needed to act effectively in recovering from the hazardous state [98].

[9] The attribution of operator error as the cause of accidents is discussed more thoroughly in *Safeware* (Chapter 5).

In the lawsuits following the 1995 Boeing-757 Cali accident (see Page 20), American Airlines was held liable for the crash based on the Colombian investigators blaming crew error entirely for the accident. The official accident investigation report cited the following four causes for the loss [2]:

1. The flightcrew's failure to adequately plan and execute the approach to runway 19 and their inadequate use of automation.
2. Failure of the flightcrew to discontinue their approach, despite numerous cues alerting them of the inadvisability of continuing the approach.
3. The lack of situational awareness of the flightcrew regarding vertical navigation, proximity to terrain, and the relative location of critical radio aids.
4. Failure of the flightcrew to revert to basic radio navigation at a time when the FMS-assisted navigation became confusing and demanded an excessive workload in a critical phase of the flight.

Look in particular the fourth identified cause: the blame is placed on the pilots when the automation became confusing and demanded an excessive workload rather than on the design of the automation. To be fair, the report also identifies two "contributory factors"—but *not* causes—as:

- FMS logic that dropped all intermediate fixes from the display(s) in the event of execution of a direct routing.
- FMS-generated navigational information that used a different naming convention from that published in navigational charts.

These two "contributory factors" are highly related to the third cause—the pilots' "lack of situational awareness." Even using an event-chain model of accidents, the FMS-related events preceded and contributed to the pilot errors. There seems to be no reason why, at the least, they should be treated any different than the labeled "causes." There were also many other factors in this accident that were not reflected in either the identified causes or contributory factors.

In this case, the Cali accident report conclusions were challenged in court. A U.S. appeals court rejected the conclusion of the report about the four causes of the accident [13], which led to a lawsuit by American in a federal court in which American Airlines alleged that components of the automated aircraft system made by Honeywell Air Transport Systems and Jeppesen Sanderson helped cause the crash. American blamed the software, saying Jeppesen stored the location of the Cali airport beacon in a different file from most other beacons. Lawyers for the computer companies argued that the beacon code could have been properly accessed and that the pilots were in error. The jury concluded that the two companies produced a defective product and that Jeppesen was 17 percent responsible, Honeywell was 8 percent at fault, and American was held to be 75 percent responsible [7]. While such distribution of responsibility may be important in determining how much each company will have to pay, it is arbitrary and does not provide any important information with respect to accident prevention in the future. The verdict is interesting, however, because the jury rejected the oversimplified notion of causality being argued. It was also one of the first cases not settled out of court where the role of software in the loss was acknowledged.

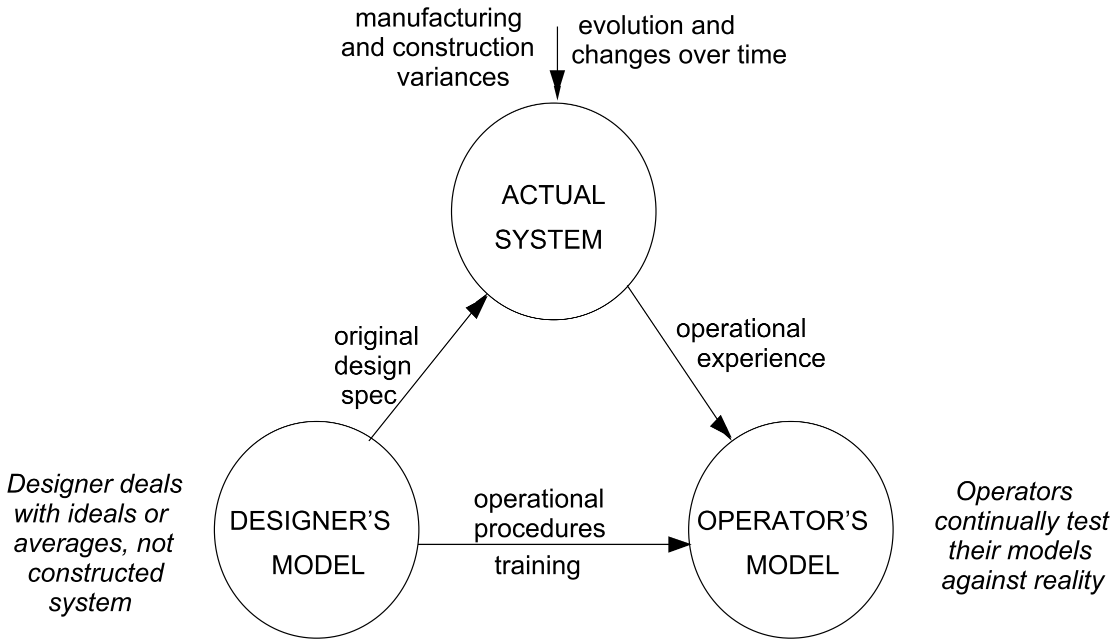

manufacturing and construction variances
evolution and changes over time
ACTUAL SYSTEM
original design spec
operational experience
Designer deals with ideals or averages, not constructed system
DESIGNER'S MODEL
operational procedures
training
OPERATOR'S MODEL
Operators continually test their models against reality

Figure 2.9: The relationship between mental models.

This case, however, does not seem to have had much impact on the attribution of pilot error in later aircraft accidents.

Part of the problem is engineers' tendency to equate people with machines. Human "failure" usually is treated the same as a physical component failure—a deviation from the performance of a specified or prescribed sequence of actions. This definition is equivalent to that of machine failure (see page 8). Alas, human behavior is much more complex than machines.

As many human factors experts have found, instructions and written procedures are almost never followed exactly as operators try to become more efficient and productive and to deal with time pressures [166]. In studies of operators, even in such highly constrained and high-risk environments as nuclear power plants, modification of instructions is repeatedly found [70, 201, 213]. When examined, these violations of rules appear to be quite rational, given the workload and timing constraints under which the operators must do their job. The explanation lies in the basic conflict between error viewed as a deviation from *normative procedure* and error viewed as a deviation from the rational and normally used *effective procedure* [168].

One implication is that following an accident, it will be easy to find someone involved in the dynamic flow of events that has violated a formal rule by following *established practice* rather than *specified practice*. Given the frequent deviation of established practice from normative work instructions and rules, it is not surprising that operator "error" is found to be the cause of 70 percent to 80 percent of accidents. As noted in the discussion of Assumption 2, a root cause is often selected because that event involves a deviation from a standard.

2.4.4 The Role of Mental Models

The updating of human mental models plays a significant role here (Figure 2.9). Both the designer and the operator will have their own mental models of the plant. It is quite natural for the designer's and operator's models to differ and even for both to have significant differences from the actual plant as it exists. During development, the designer evolves a model of the plant to the point where it can be built. The *designer's model* is an idealization formed *before* the plant is constructed. Significant differences may exist between this ideal model and the actual constructed system. Besides construction problems, the designer always deals with ideals or averages, not with the actual components themselves. Thus, a designer may have a model of a valve with an average closure time, while real valves have closure times that fall somewhere along a continuum of timing behavior that reflects manufacturing and material differences. The designer's idealized model is used to develop operator work instructions and training. But the actual system may differ from the designer's model because of manufacturing and construction variances and evolution and changes over time.

The *operator's model* of the system will be based partly on formal training created from the designer's model and partly on experience with the system. The operator must cope with the system as it is constructed and not as it may have been envisioned. As the physical system changes and evolves over time, the operator's model and operational procedures must change accordingly. While the formal procedures, work instructions, and training will be updated periodically to reflect the current operating environment, there is necessarily always a time lag. In addition, the operator may be working under time and productivity pressures that are not reflected in the idealized procedures and training.

Operators use feedback to update their mental models of the system as the system evolves. The only way for the operator to determine that the system has changed and that his or her mental model must be updated is through experimentation: To learn where the boundaries of safe behavior currently are, occasionally they must be crossed.

Experimentation is important at all levels of control [165]. For manual tasks where the optimization criteria are speed and smoothness, the limits of acceptable adaptation and optimization can only be known from the error experienced when occasionally crossing a limit. Errors are an integral part of maintaining a skill at an optimal level and a necessary part of the feedback loop to achieve this goal. The role of such experimentation in accidents cannot be understood by treating human errors as events in a causal chain separate from the feedback loops in which they operate.

At higher levels of cognitive control and supervisory decision making, experimentation is needed for operators to update procedures to handle changing conditions or to evaluate hypotheses while engaged in reasoning about the best response to unexpected situations. Actions that are quite rational and important during the search for information and test of hypotheses may appear to be unacceptable mistakes in hindsight, without access to the many details of a "turbulent" situation [168].

The ability to adapt mental models through experience in interacting with the operating system is what makes the human operator so valuable. For the reasons discussed, the operators' actual behavior may differ from the prescribed procedures because it is based on current inputs and feedback. When the deviation is correct (the designers' models are less accurate than the operators' models at that particular instant in time), then the operators are considered to be doing their job. When the operators' models are incorrect, they are often blamed for any unfortunate results, even though their incorrect mental models may have been reasonable given the information they had at the time.

Providing feedback and allowing for experimentation in system design, then, is critical in allowing operators to optimize their control ability. In the less automated system designs of the past, operators naturally had this ability to experiment and update their mental models of the current system state. Designers of highly automated systems sometimes do not understand this requirement and design automation that takes operators "out of the loop." Everyone is then surprised when the operator makes a mistake based on an incorrect mental model. Unfortunately, the reaction to such a mistake is to add even more automation and to marginalize the operators even more, thus exacerbating the problem [49].

Flawed decisions may also result from limitations in the boundaries of the operator's or designer's model. Decision makers may simply have too narrow a view of the system their decisions will impact. Recall Figure 2.2 and the discussion of the *Herald of Free Enterprise* accident. The boundaries of the system model relevant to a particular decision maker may depend on the activities of several other decision makers found within the total system [166]. Accidents may result from the interaction and side effects of their decisions based on their limited model. Before an accident, it will be difficult for the individual decision makers to see the full picture during their daily operational decision making and to judge the current state of the multiple defenses and safety margins that are partly dependent on decisions made by other people in other departments and organizations [166].

Rasmussen stresses that most decisions are sound using local judgment criteria and given the time and budget pressures and short-term incentives that shape behavior. Experts do their best to meet local conditions and in the busy daily flow of activities may be unaware of the potentially dangerous side effects of their behavior. Each individual decision may appear safe and rational within the context of the individual work environments and local pressures, but may be unsafe when considered as a whole: It is difficult—if not impossible—for any individual to judge the safety of their decisions when it is dependent on the decisions made by other people in other departments and organizations.

Decentralized decision making is, of course, required in some time-critical situations. But like all safety-critical decision making, the decentralized decisions must be made in the context of system-level information and from a total systems perspective in order to be effective in reducing accidents. One way to make distributed decision making safe is to decouple the system components in the overall system design, if possible, so that decisions do not have system-wide repercussions. Another common way to deal with the problem is to specify and train standard emergency responses. Operators may be told to sound the evacuation alarm any time an indicator reaches a certain level. In this way, safe procedures are determined at the system level and operators are socialized and trained to provide uniform and appropriate responses to crisis situations.

There are situations, of course, when unexpected conditions occur and avoiding losses requires the operators to violate the specified (and in such cases unsafe) procedures. If the operators are expected to make decisions in real time and not just follow a predetermined procedure, then they usually must have the relevant *system*-level information about the situation in order to make safe decisions. This is not true, of course, if the system design decouples the components and thus allows operators to make independent safe decisions. Such decoupling must be designed into the system, however.

Some High Reliability Organization (HRO) theorists have argued just the opposite. They have asserted that HROs are safe because they allow professionals at the front lines to use their knowledge and judgment to maintain safety. During crises, they argue, decision making in HROs migrates to the frontline workers who have the necessary judgment to make decisions [206]. The problem is that the assumption that frontline workers will have the necessary knowledge and judgment to make decisions is not necessarily true. One example is the friendly fire accident analyzed in Chapter 5 where the pilots ignored the rules of engagement they were told to follow and decided to make real-time decisions on their own based on the inadequate information they had.

Many of the HRO theories were derived from studying safety-critical systems, such as aircraft carrier flight operations. La Porte and Consolini [106], for example, argue that while the operation of aircraft carriers is subject to the Navy's chain of command, even the lowest-level seaman can abort landings. Clearly, this local authority is necessary in the case of aborted landings because decisions must be made too quickly to go up a chain of command. But note that such low-level personnel can only make decisions in one direction, that is, they may only abort landings. In essence, they are allowed to change to an inherently safe state (a go-around) with respect to the hazard involved. System-level information is not needed because a safe state exists that has no conflicts with other hazards, and the actions governed by these decisions and the conditions for making them are relatively simple. Aircraft carriers are usually operating in areas containing little traffic—they are decoupled from the larger system—and therefore localized decisions to abort are almost always safe and can be allowed from a larger system safety viewpoint.

Consider a slightly different situation, however, where a pilot makes a decision to go-around (abort a landing) at a busy urban airport. While executing a go-around when a clear danger exists if the pilot lands is obviously the right decision, there have been near misses when a pilot executed a go-around and came too close to another aircraft that was taking off on a perpendicular runway. The solution to this problem is not at the decentralized level—the individual pilot lacks the system-level information to avoid hazardous system states in this case. Instead, the solution must be at the system level, where the danger must be reduced by instituting different landing and takeoff procedures, building new runways, redistributing air traffic, or by making other system-level changes. We want pilots to be able to execute a go-around if they feel it is necessary, but unless the encompassing system is designed to prevent collisions, the action decreases one hazard while increasing a different one. Safety is a system property.

2.4.5 An Alternative View of Human Error

Traditional decision-making research views decisions as discrete processes that can be separated from the context in which the decisions are made and studied as an isolated phenomenon. This view is starting to be challenged. Instead of thinking of operations as predefined sequences of actions, human interaction with a system is increasingly being considered to be a continuous control task in which separate "decisions" or errors are difficult to identify.

Edwards, back in 1962, was one of the first to argue that decisions can only be understood as part of an ongoing process [62]. The state of the system is perceived in terms of possible actions, one of these actions is chosen, and the resulting response from the controlled system acts as a background for the next actions. Errors then are difficult to localize in the stream of behavior; the effects of less successful actions are a natural part of the search by the operator for optimal performance. As an example, consider steering a boat. The helmsman of ship A may see an obstacle ahead (perhaps another ship) and decide to steer the boat to the left to avoid it. The wind, current, and wave action may require the helmsman to make continual adjustments in order to hold the desired course. At some point, the other ship may also change course, making the helmsman's first decision about what would be a safe course no longer correct and needing to be revised. Steering then can be perceived as a continuous control activity or process with what is the correct and safe behavior changing over time and with respect to the results of prior behavior. The helmsman's mental model of the effects of the actions of the sea and the assumed behavior of the other ship has to be continually adjusted.

Not only are individual unsafe actions difficult to identify in this non-traditional control model of human decision making, but the study of decision making cannot be separated from a simultaneous study of the social context, the value system in which it takes place, and the dynamic work process it is intended to control [165]. This view is the foundation of some modern trends in decision-making research, such as *dynamic decision making* [24], the new field of *naturalistic decision making* [217, 101], and the approach to safety described in this book.

As argued by Rasmussen and others, devising more effective accident models that go beyond the simple event chain and human failure models requires shifting the emphasis in explaining the role of humans in accidents from error (that is, deviations from normative procedures) to focus instead on the mechanisms and factors that shape human behavior, that is, the performance-shaping context in which human actions take place and decisions are made. Modeling human behavior by decomposing it into decisions and actions and studying it as a phenomenon isolated from the context in which the behavior takes place is not an effective way to understand behavior [166].

The alternative view requires a new approach to representing and understanding human behavior, focused not on human error and violation of rules but on the mechanisms generating behavior in the actual, dynamic context. Such as approach must take into account the work system constraints, the boundaries of acceptable performance, the need for experimentation, and the subjective criteria guiding adaptation to change. In this approach, traditional task analysis is replaced or augmented with *cognitive work analysis* [168, 202] or *cognitive task analysis* [74]. Behavior is modeled in terms of the objectives of the decision maker, the boundaries of acceptable performance, the behavior-shaping constraints of the environment (including the value system and safety constraints), and the adaptive mechanisms of the human actors.

Such an approach leads to new ways of dealing with the human contribution to accidents and human "error." Instead of trying to control human behavior by fighting deviations from specified procedures, focus should be on controlling behavior by identifying the boundaries of safe performance (the behavioral safety constraints), by making the boundaries explicit and known, by giving opportunities to develop coping skills at the boundaries, by designing systems to support safe optimization and adaptation of performance in response to contextual influences and pressures, by providing means for identifying potentially dangerous side effects of individual decisions in the network of decisions over the entire system, by designing for error tolerance (making errors observable and reversible before safety constraints are violated) [166], and by counteracting the pressures that drive operators and decision makers to violate safety constraints.

Once again, future progress in accident reduction requires tossing out the old assumption and substituting a new one:

New Assumption 4: Operator behavior is a product of the environment in which it occurs. To reduce operator "error" we must change the environment in which the operator works.

Human behavior is always influenced by the environment in which it takes place. Changing that environment will be much more effective in changing operator error than the usual behaviorist approach of using reward and punishment. Without changing the environment, human error cannot be reduced for long. We design systems in which operator error is inevitable and then blame the operator and not the system design.

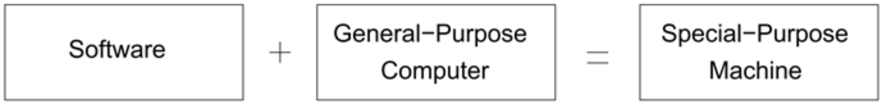
As argued by Rasmussen and others, devising more effective accident causality models requires shifting the emphasis in explaining the role that humans play in accidents from error (deviations from normative procedures) to focus on the mechanisms and factors that shape human behavior, that is, the performance-shaping features and context in which human actions take place and decisions are made. Modeling behavior by decomposing it into decisions and actions or events, which most all current accident models do, and studying it as a phenomenon isolated from the context in which the behavior takes place is not an effective way to understand behavior [166].

2.5 The Role of Software in Accidents

Assumption 5: Highly reliable software is safe.

The most common approach to ensuring safety when the system includes software is to try to make the software highly reliable. To help readers who are not software professionals see the flaws in this assumption, a few words about software in general may be helpful.

The uniqueness and power of the digital computer over other machines stems from the fact that, for the first time, we have a general-purpose machine:

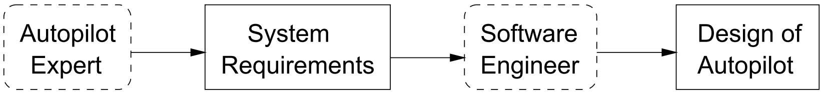


We no longer need to build a mechanical or analog autopilot from scratch, for example, but simply to write down the "design" of an autopilot in the form of instructions or steps to accomplish the desired goals. These steps are then loaded into the computer, which, while executing the instructions, in effect *becomes* the special-purpose machine (the autopilot). If changes are needed, the instructions can be changed and the same physical machine (the computer hardware) is used instead of having to build a different physical machine from scratch. Software in essence is the *design of a machine abstracted from its physical realization.* In other words, the logical design of a machine (the software) is separated from the physical design of that machine (the computer hardware).

Machines that previously were physically impossible or impractical to build become feasible, and the design of a machine can be changed quickly without going through an entire retooling and manufacturing process. In essence, the manufacturing phase is eliminated from the lifecycle of these machines: the physical parts of the machine (the computer hardware) can be reused, leaving only the design and verification phases. The design phase also has changed: The designer can concentrate on identifying the steps to be achieved without having to worry about how those steps will be realized physically.

These advantages of using computers (along with others specific to particular applications, such as reduced size and weight) have led to an explosive increase in their use, including their introduction into potentially dangerous systems. There are, however, some potential disadvantages of using computers and some important changes that their use introduces into the traditional engineering process that are leading to new types of accidents as well as creating difficulties in investigating accidents and preventing them.

One of the most important changes is that with computers, the design of the special purpose machine is usually created by someone who is not an expert on designing such machines. The autopilot design expert, for example, decides how the autopilot should work, and then provides that information to a software engineer, who is an expert in software design but not autopilots. It is the software engineer who then creates the detailed design of the autopilot. The extra communication step between the engineer and the software developer is the source of the most serious problems with software today.



It should not be surprising, then, that most errors found in operational software can be traced to requirements flaws, particularly incompleteness. Completeness is a quality often associated with requirements but rarely defined. The most appropriate definition in the context of this book has been proposed by Jaffe: Software requirements specifications are complete if they are sufficient to distinguish the desired behavior of the software from that of any other undesired program that might be designed [90].

Nearly all the serious accidents in which software has been involved in the past twenty years can be traced to requirements flaws, not coding errors. The requirements may reflect incomplete or wrong assumptions

- About the operation of the system components being controlled by the software (for example, how quickly the component can react to a software-generated control command) or

- About the required operation of the computer itself.

. In the Mars Polar Lander loss the software requirements did not include information about the potential for the landing leg sensors to generate noise or, alternatively, to ignore any inputs from the sensors while the spacecraft was more than forty meters above the planet surface. In the batch chemical reactor accident, the software engineers were never told to open the water valve before the catalyst valve and apparently thought the ordering was therefore irrelevant.

The problems may also stem from unhandled controlled-system states and environmental conditions. An F-18 was lost when a mechanical failure in the aircraft led to the inputs arriving faster than expected, which overwhelmed the software [69]. Another F-18 loss resulted from the aircraft getting into an attitude that the engineers had assumed was impossible and that the software was not programmed to handle.

In these cases, simply trying to get the software "correct" in terms of accurately implementing the requirements will not make it safer. Software may be highly reliable and correct and still be unsafe when:

- The software correctly implements the requirements but the specified behavior is unsafe from a system perspective;
- The software requirements do not specify some particular behavior required for system safety (that is, they are incomplete);
- The software has unintended (and unsafe) behavior beyond what is specified in the requirements.

If the problems stem from the software doing what the software engineer thought it should do when that is not what the original design engineer wanted, the use of integrated product teams and other project management schemes to help with communication are useful. The most serious problems arise, however, when *nobody* understands what the software should do or even what it should not do. We need better techniques to assist in determining these requirements.

There is not only anecdotal but some hard data to support the hypothesis that safety problems in software stem from requirements not coding errors. Lutz examined 387 software errors uncovered during integration and system testing of the Voyager and Galileo spacecraft [129]. She concluded that the software errors identified as potentially hazardous to the system tended to be produced by different error mechanisms than non-safety-related software errors. She showed that for these two spacecraft, the safety-related software errors arose most commonly from (1) discrepancies between the documented requirements specifications and the requirements needed for correct functioning of the system and (2) misunderstandings about the software's interface with the rest of the system. They did not involve coding errors in implementing the documented requirements.

Many software requirements problems arise from what could be called the *Curse of Flexibility.* The computer is so powerful and so useful because it has eliminated many of the physical constraints of previous machines. This is both its blessing and its curse: We no longer have to worry about the physical realization of our designs, but we also no longer have physical laws that limit the complexity of our designs. Physical constraints enforce discipline on the design, construction, and modification of our design artifacts. Physical constraints also control the complexity of what we build. With software, the limits of what is *possible* to accomplish are different than the limits of what can be accomplished *successfully* and *safely*—the limiting factors change from the structural integrity and physical constraints of our materials to limits on our intellectual capabilities.

It is possible and even quite easy to build software that we cannot understand in terms of being able to determine how it will behave under all conditions. We can construct software (and often do) that goes beyond human intellectual limits. The result has been an increase in component interaction accidents stemming from intellectual unmanageability that allows potentially unsafe interactions to go undetected during development. The software often controls the interactions among the system components so its close relationship with component interaction accidents should not be surprising. But this fact has important implications for how software must be engineered when it controls potentially unsafe systems or products: Software or system engineering techniques that simply ensure software reliability or correctness (consistency of the code with the requirements) will have little or no impact on safety.

Techniques that *are* effective will rest on a new assumption:

New Assumption 5: Highly reliable software is not necessarily safe. Increasing software reliability or reducing implementation errors will have little impact on safety.

2.6 Static versus Dynamic Views of Systems

Assumption 6: Major accidents occur from the chance simultaneous occurrence of random events.

Most current safety engineering techniques suffer from the limitation of considering only the events underlying an accident and not the entire accident *process*. Accidents are often viewed as some unfortunate coincidence of factors that come together at one particular point in time and lead to the loss. This belief arises from too narrow a view of the causal time line. Looking only at the immediate time of the Bhopal MIC release, it does seem to be a coincidence that the refrigeration system, flare tower, vent scrubber, alarms, water curtain, and so on, had all been inoperable at the same time. But viewing the accident through a larger lens makes it clear that the causal factors were all related to systemic causes that had existed for a long time.

Systems are not static. Rather than accidents being a chance occurrence of multiple independent events, they tend to involve a migration to a state of increasing risk over time [166]. A point is reached where an accident is inevitable unless the high risk is detected and reduced. The particular events involved at the time of the loss are somewhat irrelevant: if those events had not occurred, something else would have led to the loss. This concept is reflected in the common observation that a loss was "an accident waiting to happen." The proximate cause of the Columbia Space Shuttle loss was the foam coming loose from the external tank and damaging the re-entry heat control structure. But many potential problems that could have caused the loss of the Shuttle had preceded this event and an accident was avoided by luck or unusual circumstances. The economic and political pressures led the Shuttle program to migrate to a state where any slight deviation could have led to a loss [116].

Any approach to enhancing safety that includes the social system and humans must account for adaptation. To paraphrase a familiar saying, the only constant is that nothing ever remains constant. Systems and organizations continually experience change as adaptations are made in response to local pressures and short-term productivity and cost goals. People adapt to their environment or they change their environment to better suit their purposes. A corollary to this propensity for systems and people to adapt over time is that safety defenses are likely to degenerate systematically through time, particularly when pressure toward cost-effectiveness and increased productivity is the dominant element in decision making. Rasmussen noted that the critical factor here is that such adaptation is not a random process—it is an optimization process depending on search strategies—and thus should be predictable and potentially controllable [166].

Woods has stressed the importance of adaptation in accidents. He describes organizational and human failures as breakdowns in adaptations directed at coping with complexity, and accidents as involving a "drift toward failure as planned defenses erode in the face of production pressures and change" [214].

Similarly, Rasmussen has argued that major accidents are often caused not by a coincidence of independent failures but instead reflect a systematic migration of organizational behavior to the boundaries of safe behavior under pressure toward cost-effectiveness in an aggressive, competitive environment [166]. One implication of this viewpoint is that the struggle for a good safety culture will never end because it must continually fight against the functional pressures of the work environment. Improvement of the safety culture will therefore require an analytical approach directed toward the behavior-shaping factors in the environment. A way of achieving this goal is described in Part III.

Humans and organizations can adapt and still maintain safety as long as they stay within the area bounded by safety constraints. But in the search for optimal operations, humans and organizations will close in on and explore the boundaries of established practice. Such exploration implies the risk of occasionally crossing the limits of safe practice unless the constraints on safe behavior are enforced.

The natural migration toward the boundaries of safe behavior, according to Rasmussen, is complicated by the fact that it results from the decisions of multiple people, in different work environments and contexts within the overall sociotechnical system, all subject to competitive or budgetary stresses and each trying to optimize their decisions within their own immediate context. Several decision makers at different times, in different parts of the company or organization, all striving locally to optimize cost effectiveness may be preparing the stage for an accident, as illustrated by the Zeebrugge ferry accident (see Figure 2.2) and the friendly fire accident described in Chapter 5. The dynamic flow of events can then be released by a single act.

Our new assumption is therefore:

New Assumption 6: Systems will tend to migrate toward states of higher risk. Such migration is predictable and can be prevented by appropriate system design or detected during operations using leading indicators of increasing risk.

To handle system adaptation over time, our causal models and safety techniques must consider the *processes* involved in accidents and not simply events and conditions: Processes control a sequence of events and describe system and human behavior as it changes and adapts over time rather than considering individual events and human actions. To talk about the cause or causes of an accident makes no sense in this systems or process view of accidents. As Rasmussen argues, deterministic causal models are inadequate to explain the organizational and social factors in highly adaptive sociotechnical systems. Instead, accident causation must be viewed as a complex process involving the entire sociotechnical system including legislators, government agencies, industry associations and insurance companies, company management, technical and engineering personnel, operations, and so on [166].

2.7 The Focus on Determining Blame

Assumption 7: Assigning blame is necessary to learn from and prevent accidents or incidents.

Beyond the tendency to blame operators described under Assumption 3, other types of subjectivity in ascribing cause exist. Rarely are all the causes of an accident perceived identically by everyone involved including engineers, managers, operators, union officials, insurers, lawyers, politicians, the press, the state, and the victims and their families. Such conflicts are typical in situations that involve normative, ethical, and political considerations about which people may legitimately disagree. Some conditions may be considered unnecessarily hazardous by one group yet adequately safe and necessary by another. In addition, judgments about the cause of an accident may be affected by the threat of litigation or by conflicting interests.

Research data validates this hypothesis. Various studies have found the selection of a cause(s) depends on characteristics of the victim and of the analyst (e.g., hierarchical status, degree of involvement, and job satisfaction) as well as on the relationships between the victim and the analyst and on the severity of the accident [111].

For example, one study found that workers who were satisfied with their jobs and who were integrated into and participating in the enterprise attributed accidents mainly to personal causes. In contrast, workers who were not satisfied and who had a low degree of integration and participation more often cited nonpersonal causes that implied that the enterprise was responsible [111]. Another study found differences in the attribution of accident causes among victims, safety managers, and general managers. Other researchers have suggested that accidents are attributed to factors in which the individuals are less directly involved. A further consideration may be position in the organization: The lower the position in the hierarchy, the greater the tendency to blame accidents on factors linked to the organization; individuals who have a high position in the hierarchy tend to blame workers for accidents [111].

There even seem to be differences in causal attribution between accidents and incidents: Accident investigation data on near-miss (incident) reporting suggest that causes for these events are mainly attributed to technical deviations while similar events that result in losses are more often blamed on operator error [61, 99].

Causal identification may also be influenced by the data collection methods. Data are usually collected in the form of textual descriptions of the sequence of events of the accident, which, as we have seen, tend to concentrate on obvious conditions or events closely preceding the accident in time and tend to leave out less obvious or indirect events and factors. There is no simple solution to this inherent bias: On one hand, report forms that do not specifically ask for nonproximal factors often do not elicit them while, on the other hand, more directive report forms that do request particular information may limit the categories or conditions considered [100].

Other factors affecting causal filtering in accident and incident reports may be related to the design of the reporting system itself. For example, the NASA Aviation Safety Reporting System (ASRS) has a category that includes non-adherence to FARs (Federal Aviation Regulations). In a NASA study of reported helicopter incidents and accidents over a nine-year period, this category was by far the largest category cited [80]. The NASA study concluded that the predominance of FAR violations in the incident data may reflect the motivation of the ASRS reporters to obtain immunity from perceived or real violations of FARs and not necessarily the true percentages.

A final complication is that human actions always involve some interpretation of the person's goals and motives. The individuals involved may be unaware of their actual goals and motivation or may be subject to various types of pressures to reinterpret their actions. Explanations by accident analysts after the fact may be influenced by their own mental models or additional goals and pressures.

Note the difference between an explanation based on goals and one based on motives: a goal represents an end state while a motive explains *why* that end state was chosen. Consider the hypothetical case where a car is driven too fast during a snowstorm and it slides into a telephone pole. An explanation based on goals for this chain of events might include the fact that the driver wanted to get home quickly. An explanation based on motives might include the fact that guests were coming for dinner and the driver had to prepare the food before they arrived.

Explanations based on goals and motives depend on assumptions that cannot be directly measured or observed by the accident investigator. Leplat illustrates this dilemma by describing three different motives for the event *"operator sweeps the floor"*: (1) the floor is dirty, (2) the supervisor is present, or (3) the machine is broken and the operator needs to find other work [112]. Even if the people involved survive the accident, true goals and motives may not be revealed for a variety of reasons.

Where does all this leave us? There are two basic reasons for conducting an accident investigation: (1) to assign blame for the accident and (2) to understand why it happened so that future accidents can be prevented. When the goal is to assign blame, the backward chain of events considered often stops when someone or something appropriate to blame is found, such as the baggage handler in the DC-10 case or the maintenance worker at Bhopal. As a result, the selected initiating event may provide too superficial an explanation of why the accident occurred to prevent similar losses in the future.

As another example, stopping at the O-ring failure in the *Challenger* accident and fixing that particular design flaw would not have eliminated the systemic flaws that could lead to accidents in the future. For *Challenger*, examples of those systemic problems include flawed decision making and the political and economic pressures that led to it, poor problem reporting, lack of trend analysis, a "silent" or ineffective safety program, communication problems, etc. None of these are "events" (although they may be manifested in particular events) and thus do not appear in the chain of events leading to the accident. Wisely, the authors of the *Challenger* accident report used an event chain only to identify the proximate physical cause and not the reasons those events occurred, and the report's recommendations led to many important changes at NASA or at least attempts to make such changes.

Twenty years later, another Space Shuttle was lost. While the proximate cause for the *Columbia* accident (foam hitting the wing of the orbiter) was very different than that for *Challenger*, many of the systemic causal factors were similar and reflected either inadequate fixes of these factors after the *Challenger* accident or their re-emergence in the years between these losses [116].

Blame is not an engineering concept; it is a legal or moral one. Usually there is no objective criterion for distinguishing one factor or several factors from other factors that contribute to an accident. While lawyers and insurers recognize that many factors contribute to a loss event, for practical reasons and particularly for establishing liability, they often oversimplify the causes of accidents and identify what they call the *proximate* (immediate or direct) cause. The goal is to determine the parties in a dispute that have the legal liability to pay damages, which may be affected by the ability to pay or by public policy considerations, such as discouraging company management or even an entire industry from acting in a particular way in the future.

When learning how to engineer safer systems is the goal rather than identifying who to punish and establishing liability, then the emphasis in accident analysis needs to shift from *cause* (in terms of events or errors), which has a limiting, blame orientation, to understanding accidents in terms of *reasons*, that is, why the events and errors occurred. In an analysis by the author of recent aerospace accidents involving software, most of the reports stopped after assigning blame—usually to the operators who interacted with the software—and never got to the root of why the accident occurred, e.g., why the operators made the errors they did and how to prevent such errors in the future (perhaps by changing the software) or why the software requirements specified unsafe behavior, why that requirements error was introduced and why it was not detected and fixed before the software was used [115].

When trying to understand operator contributions to accidents, just as with overcoming hindsight bias, it is more helpful in learning how to prevent future accidents by focusing *not* on what the operator did "wrong" but on why it made sense for them to behave that way under those conditions [50]. Most people are not malicious, but are simply trying to do the best they can under the circumstances and with the information they have. Understanding why those efforts were not enough will help in changing features of the system and environment so that sincere efforts are more successful in the future. Focusing on assigning blame contributes nothing toward achieving this goal and may impede it by reducing openness during accident investigations, thereby making it more difficult to find out what really happened.

A focus on blame can also lead to a lot of finger pointing and arguments that someone or something else was more to blame. Much effort is usually spent in accident investigations on determining which factors were the most important and assigning them to categories such as root cause, primary cause, contributory cause. In general, determining the relative importance of various factors to an accident may not be useful in preventing future accidents. Haddon [76] argues, reasonably, that countermeasures to accidents should *not* be determined by the relative importance of the causal factors; instead, priority should be given to the measures that will be most effective in reducing future losses. Explanations involving events in an event chain often do not provide the information necessary to prevent future losses, and spending a lot of time determining the relative contributions of events or conditions to accidents (such as arguing about whether an event is the root cause or a contributory cause) is not productive outside the legal system. Rather, Haddon suggests that engineering effort should be devoted to identifying the factors (1) that are easiest or most feasible to change, (2) that will prevent large classes of accidents, and (3) over which we have the greatest control.

Because the goal of this book is to describe a new approach to understanding and preventing accidents rather than assigning blame, the emphasis is on identifying *all* the factors involved in an accident and understanding the relationship among these causal factors in order to provide an explanation of why the accident occurred. That explanation can then be used to generate recommendations for preventing losses in the future. Building safer systems will be more effective when we consider all causal factors, both direct and indirect. In the new approach presented in this book, there is no attempt to determine which factors are more "important" than others but rather how they all relate to each other and to the final loss event or near miss.

One final new assumption is needed to complete the foundation for future progress:

New Assumption 7: *Blame is the enemy of safety. Focus should be on understanding how the system behavior as a whole contributed to the loss and not on who or what to blame for it.*

We will be more successful in enhancing safety by focusing on why accidents occur rather than on blame.

Updating our assumptions about accident causation will allow us to make greater progress toward building safer systems in the twenty-first century. The old and new assumptions are summarized in Table 1. The new assumptions provide the foundation for a new view of accident causation.

Table 1: The Basis for a New Foundation for Safety Engineering

Old Assumption	**New Assumption**
Safety is increased by increasing system or component reliability; if components do not fail, then accidents will not occur.	High reliability is neither necessary nor sufficient for safety.
Accidents are caused by chains of directly related events. We can understand accidents and assess risk by looking at the chains of events leading to the loss.	Accidents are complex processes involving the entire sociotechnical system. Traditional event-chain models cannot describe this process adequately
Probabilistic risk analysis based on event chains is the best way to assess and communicate safety and risk information.	Risk and safety may be best understood and communicated in ways other than probabilistic risk analysis.
Most accidents are caused by operator error. Rewarding safe behavior and punishing unsafe behavior will eliminate or reduce accidents significantly.	Operator error is a product of the environment in which it occurs. To reduce operator "error" we must change the environment in which the operator works.
Highly reliable software is safe.	Highly reliable software is not necessarily safe. Increasing software reliability will have only minimal impact on safety.
Major accidents occur from the chance simultaneous occurrence of random events.	Systems will tend to migrate toward states of higher risk. Such migration is predictable and can be prevented by appropriate system design or detected during operations using leading indicators of increasing risk.
Assigning blame is necessary to learn from and prevent accidents or incidents.	Blame is the enemy of safety. Focus should be on understanding how the system behavior as a whole contributed to the loss and not on who or what to blame for it.

2.8 Goals for a New Accident Model

Event-based models work best for accidents where one or several components fail, leading to a system failure or hazard. Accident models and explanations involving only simple chains of failure events, however, can easily miss subtle and complex couplings and interactions among failure events and omit entirely accidents involving no component failure at all. The event-based models developed to explain physical phenomena (which they do well) are inadequate to explain accidents involving organizational and social factors and human decisions and software design errors in highly adaptive, tightly-coupled, interactively complex sociotechnical systems—namely, those accidents related to the new factors (described in Chapter 1) in the changing environment in which engineering is taking place.

The search for a new model, resulting in the accident model presented in Part II of this book, was driven by the following goals:

- *Expand accident analysis by forcing consideration of factors other than component failures and human errors.* The model should encourage a broad view of accident mechanisms, expanding the investigation from simply considering proximal events to considering the entire sociotechnical system. Such a model should include societal, regulatory, and cultural factors. While some accident reports do this well, for example the space shuttle *Challenger* report, such results appear to be ad hoc and dependent on the personalities involved in the investigation rather than being guided by the accident model itself.

- *Provide a more scientific way to model accidents that produces a better and less subjective understanding of why the accident occurred and how to prevent future ones.* Event-chain models provide little guidance in the selection of events to include in the accident explanation or the conditions to investigate. The model should provide more assistance in identifying and understanding a comprehensive set of factors involved, including the adaptations that led to the loss.

- *Include system design errors and dysfunctional system interactions.* The models used widely were created before computers and digital components and do not handle them well. In fact, many of the event-based models were developed to explain industrial accidents, such as workers falling into holes or injuring themselves during the manufacturing process, and do not fit system safety at all. A new model must be able to account for accidents arising from dysfunctional interactions among the system components.

- *Allow for and encourage new types of hazard analyses and risk assessments that go beyond component failures and can deal with the complex role software and humans are assuming in high-tech systems.* Traditional hazard analysis techniques, such as fault tree analysis and the various other types of failure analysis techniques, do not work well for human errors and for software and other system design errors. An appropriate model should suggest hazard analysis techniques to augment these failure-based methods and encourage a wider variety of risk reduction measures than redundancy and monitoring. In addition, risk assessment is currently firmly rooted in the probabilistic analysis of failure events. Attempts to extend current probabilistic risk assessment techniques to software and other new technology, to management, and to cognitively complex human control activities have been disappointing. This way forward may lead to a dead end, but starting from a different theoretical foundation may allow significant progress in finding new, more comprehensive approaches to risk assessment for complex systems.

- *Shift the emphasis in the role of humans in accidents from errors (deviations from normative behavior) to focus on the mechanisms and factors that shape human behavior (i.e., the performance-shaping mechanisms and context in which human actions take place and decisions are made).* A new model should account for the complex role that human decisions and behavior are playing in the accidents occurring in high-tech systems and handle not simply individual decisions but also sequences of decisions and the interactions among decisions by multiple, interacting decision makers [166]. The model must include examining the possible goals and motives behind human behavior as well as the contextual factors that influenced that behavior.

- *Encourage a shift in the emphasis in accident analysis from "cause"—which has a limiting,* *blame orientation—to understanding accidents in terms of reasons, that is, why the events and errors occurred [196].* Learning how to engineer safer systems is the goal here, not identifying whom to punish.

- *Examine the processes involved in accidents and not simply events and conditions.* Processes control a sequence of events and describe changes and adaptations over time rather than considering events and human actions individually.

- *Allow for and encourage multiple viewpoints and multiple interpretations when appropriate.* Operators, managers, and regulatory agencies may all have different views of the flawed processes underlying an accident, depending on the hierarchical level of the sociotechnical control structure from which the process is viewed. At the same time, the factual data should be separated from the interpretation of that data.

- *Assist in defining operational metrics and analyzing performance data.* Computers allow the collection of massive amounts of operational data, but analyzing that data to determine whether the system is moving toward the boundaries of safe behavior is difficult. A new accident model should provide directions for identifying appropriate safety metrics and operational auditing procedures to evaluate decisions made during design and development, to determine whether controls over hazards are adequate, to detect erroneous operational and environmental assumptions underlying the hazard analysis and design process, to identify leading indicators and dangerous trends and changes in operations before they lead to accidents, and to identify any maladaptive system or environment changes over time that could increase accident risk to unacceptable levels.

These goals are achievable if models based on systems theory, rather than reliability theory, underlie our safety engineering activities.

Chapter 3

Systems Theory and Its Relationship to Safety

To achieve the goals set at the end of the last chapter, a new theoretical underpinning is needed for system safety. Systems theory provides that foundation. This chapter introduces some basic concepts in systems theory, how this theory is reflected in system engineering, and how all of this relates to system safety.

3.1 An Introduction to Systems Theory

Systems theory dates from the 1930s and 1940s and was a response to limitations of the classic analysis techniques in coping with the increasingly complex systems starting to be built at that time [35]. Norbert Wiener applied the approach to control and communications engineering [210] while Ludwig von Bertalanffy developed similar ideas for biology [20]. Bertalanffy suggested that the emerging ideas in various fields could be combined into a general theory of systems.

In the traditional scientific method, sometimes referred to as *divide and conquer*, systems are broken into distinct parts so that the parts can be examined separately: Physical aspects of systems are decomposed into separate physical components while behavior is decomposed into discrete events over time.

$$\begin{array}{rl} \text{Physical aspects} & \Longrightarrow \text{Separate physical components} \\ \text{Behavior} & \Longrightarrow \text{Discrete events over time} \end{array}$$

This decomposition (formally called *analytic reduction*) assumes that the separation is feasible: that is, each component or subsystem operates independently and analysis results are not distorted when these components are considered separately. This assumption in turn implies that the components or events are not subject to feedback loops and other nonlinear interactions and that the behavior of the components is the same when examined singly as when they are playing their part in the whole. A third fundamental assumption is that the principles governing the assembling of the components into the whole are straightforward, that is, the interactions among the subsystems are simple enough that they can be considered separate from the behavior of the subsystems themselves.

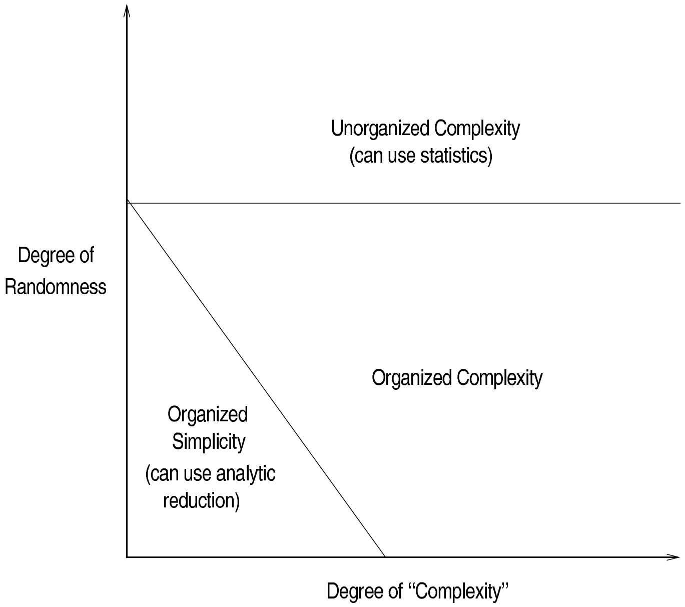


Figure 3.1: Three Categories of Systems (Adapted from Gerald Weinberg, *An Introduction to General Systems Thinking*, John Wiley, 1975)

These are reasonable assumptions, it turns out, for many of the physical regularities of the universe. System theorists have described these systems as displaying *organized simplicity* (Figure 3.1) [207]. Such systems can be separated into non-interacting subsystems for analysis purposes: the precise nature of the component interactions is known and interactions can be examined pairwise. Analytic reduction has been highly effective in physics and is embodied in structural mechanics.

Other types of systems display what systems theorists have labeled *unorganized complexity*—that is, they lack the underlying structure that allows reductionism to be effective. They can, however, often be treated as aggregates: They are complex but regular and random enough in their behavior that they can be studied statistically. This study is simplified by treating them as a structureless mass with interchangeable parts and then describing them in terms of averages. The basis of this approach is the *law of large numbers*: The larger the population, the more likely that observed values are close to the predicted average values. In physics, this approach is embodied in statistical mechanics.

A third type of system exhibits what system theorists call *organized complexity*: These systems are too complex for complete analysis and too organized for statistics; the averages are deranged by the underlying structure [207]. Many of the complex engineered systems of the post-World War II era, as well as biological systems and social systems, fit into this category. Organized complexity also represents particularly well the problems that are faced by those attempting to build complex software, and it explains the difficulty computer scientists have had in attempting to apply analysis and statistics to software.

Systems theory was developed for this third type of system. The systems approach focuses on systems taken as a whole, not on the parts taken separately. It assumes that some properties of systems can only be treated adequately in their entirety, taking into account all facets relating the social to the technical aspects [160]. These system properties derive from the relationships between the parts of systems: how the parts interact and fit together [1]. Concentrating on the analysis and design of the whole as distinct from the components or parts provides a means for studying systems exhibiting organized complexity.

The foundation of systems theory rests on two pairs of ideas: (1) *emergence* and *hierarchy* and (2) *communication* and *control* [35].

3.2 Emergence and Hierarchy

A general model of complex systems can be expressed in terms of a *hierarchy* of levels of organization, each more complex than the one below, where a level is characterized by having *emergent* properties. Emergent properties do not exist at lower levels; they are meaningless in the language appropriate to those levels. The shape of an apple, although eventually explainable in terms of the cells of the apple, has no meaning at that lower level of description. The operation of the processes at the lower levels of the hierarchy result in a higher level of complexity—that of the whole apple itself—that has emergent properties, one of them being the apple's shape [35]. The concept of emergence is the idea that at a given level of complexity, some properties characteristic of that level (emergent at that level) are irreducible.

Hierarchy theory deals with the fundamental differences between one level of complexity and another. Its ultimate aim is to explain the relationships between different levels: what generates the levels, what separates them, and what links them. Emergent properties associated with a set of components at one level in a hierarchy are related to *constraints upon the degree of freedom* of those components. Describing the emergent properties resulting from the imposition of constraints requires a language at a higher level (a metalevel) different than that describing the components themselves. Thus, different languages of description are appropriate at different levels.

Reliability is a component property.[1] Conclusions can be reached about the reliability of a valve in isolation, where reliability is defined as the probability that the behavior of the valve will satisfy its specification over time and under given conditions.

Safety, on the other hand, is clearly an emergent property of systems: Safety can only be determined in the context of the whole. Determining whether a plant is acceptably safe is not possible, for example, by examining a single valve in the plant. In fact, statements about the "safety of the valve" without information about the context in which that valve is used, are meaningless. Safety is determined by the relationship between the valve and the other plant components. As another example, pilot procedures to execute a landing might be safe in one aircraft or in one set of circumstances but unsafe in another.

While often confused, reliability and safety are different properties. The pilots may reliably execute the landing procedures on a plane or at an airport in which those procedures are unsafe. A gun when discharged out on a desert with no other humans or animals for hundreds of miles may be both safe and reliable. When discharged in a crowded mall, the reliability will not have changed but the safety most assuredly has.

Because safety is an emergent property, it is not possible to take a single system component, like a software module or a single human action, in isolation and assess its safety. A component

[1] This statement is somewhat of an oversimplification as the reliability of a system component can, under some conditions (e.g., magnetic interference or excessive heat) be impacted by its environment. The basic reliability of the component, however, can be defined and measured in isolation whereas the safety of an individual component is undefined except in a specific environment.

that is perfectly safe in one system or in one environment may not be when used in another.

The new model of accidents introduced in Part II of this book incorporates the basic systems theory idea of hierarchical levels, where constraints or lack of constraints at the higher levels control or allow lower-level behavior. Safety is treated as an emergent property at each of these levels. Safety depends on the enforcement of constraints on the behavior of the components in the system, including constraints on their potential interactions. Safety in the batch chemical reactor in the previous chapter, for example, depends on the enforcement of a constraint on the relationship between the state of the catalyst valve and the water valve.

3.3 Communication and Control

The second major pair of ideas in systems theory is *communication* and *control*. An example of regulatory or *control* action is the imposition of *constraints* upon the activity at one level of a hierarchy, which define the "laws of behavior" at that level. Those laws of behavior yield activity meaningful at a higher level. Hierarchies are characterized by control processes operating at the interfaces between levels [35].

The link between control mechanisms studied in natural systems and those engineered in man-made systems was provided by a part of systems theory known as cybernetics. Checkland writes:

> Control is always associated with the imposition of constraints, and an account of a control process necessarily requires our taking into account at least two hierarchical levels. At a given level, it is often possible to describe the level by writing dynamical equations, on the assumption that one particle is representative of the collection and that the forces at other levels do not interfere. But any description of a control process entails an upper level imposing constraints upon the lower. The upper level is a source of an alternative (simpler) description of the lower level in terms of specific functions that are emergent as a result of the imposition of constraints [35, p.87].

Note Checkland's statement about control always being associated with the imposition of constraints. Imposing *safety constraints* plays a fundamental role in the approach to safety presented in this book. The limited focus on avoiding failures, which is common in safety engineering today, is replaced by the larger concept of imposing constraints on system behavior to avoid unsafe events or conditions, that is, hazards.

Control in open systems (those that have inputs and outputs from their environment) implies the need for *communication*. Bertalanffy distinguished between *closed systems*, in which unchanging components settle into a state of equilibrium, and *open systems*, which can be thrown out of equilibrium by exchanges with their environment.

In control theory, open systems are viewed as interrelated components that are kept in a state of dynamic equilibrium by feedback loops of information and control. The plant's overall performance has to be controlled in order to produce the desired product while satisfying cost, safety, and general quality constraints.

In order to control a process, four conditions are required [10]:

- **Goal Condition**: The controller must have a goal or goals (for example, to maintain the setpoint).

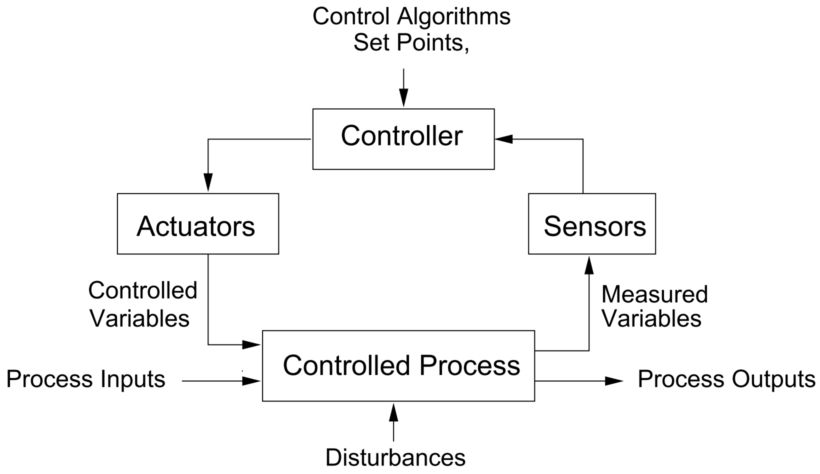


Figure 3.2: A standard control loop.

- **Action Condition**: The controller must be able to affect the state of the system. In engineering, control actions are implemented by *actuators.*
- **Model Condition**: The controller must be (or contain) a model of the system (see Section 4.3).
- **Observability Condition**: The controller must be able to ascertain the state of the system. In engineering terminology, observation of the state of the system is provided by *sensors.*

Figure 3.2 shows a typical control loop. The plant controller obtains information about (observes) the process state from measured variables (*feedback*) and uses this information to initiate action by manipulating *controlled variables* to keep the process operating within predefined limits or *set points* (the goal) despite disturbances to the process. In general, the maintenance of any open-system hierarchy (either biological or man-made) will require a set of processes in which there is communication of information for regulation or control [35].

Control actions will generally lag in their effects on the process because of delays in signal propagation around the control loop: an actuator may not respond immediately to an external command signal (called *dead time*); the process may have delays in responding to manipulated variables (*time constants*); and the sensors may obtain values only at certain sampling intervals (*feedback delays*). Time lags restrict the speed and extent with which the effects of disturbances, both within the process itself and externally derived, can be reduced. They also impose extra requirements on the controller, for example, the need to infer delays that are not directly observable.

The model condition plays an important role in accidents and safety. In order to create effective control actions, the controller must know the current state of the controlled process and be able to estimate the effect of various control actions on that state. As discussed further in section 4.3, many accidents have been caused by the controller incorrectly assuming the controlled system was in a particular state and imposing a control action (or not providing one) that led to a loss: the Mars Polar Lander descent engine controller, for example, assumed the spacecraft was on the surface of the planet and shut down the descent engines. The captain of the Herald of Free Enterprise thought the car deck doors were shut and left the mooring.

3.4 Using System Theory to Understand Accidents

Safety approaches based on system theory consider accidents as arising from the interactions among system components and usually do not specify single causal variables or factors [111]. Whereas industrial (occupational) safety models and event chain models focus on unsafe acts or conditions, classic system safety models instead look at what went wrong with the system's operation or organization to allow the accident to take place.

This systems approach treats safety as an emergent property that arises when the system components interact within an environment. Emergent properties like safety are controlled or enforced by a set of constraints (control laws) related to the behavior of the system components. For example, the spacecraft descent engines must remain on until the spacecraft reaches the surface of the planet and the car deck doors on the ferry must be closed before leaving port. Accidents result from interactions among components that violate these constraints—in other words, from a lack of appropriate constraints on the interactions. Component interaction accidents, as well as component failure accidents, can be explained using these concepts.

Safety then can be viewed as a control problem. Accidents occur when component failures, external disturbances, and/or dysfunctional interactions among system components are not adequately controlled. In the space shuttle *Challenger* loss, the O-rings did not adequately control propellant gas release by sealing a tiny gap in the field joint. In the Mars Polar Lander loss, the software did not adequately control the descent speed of the spacecraft—it misinterpreted noise from a Hall effect sensor (feedback of a measured variable) as an indication the spacecraft had reached the surface of the planet. Accidents such as these, involving engineering design errors, may in turn stem from inadequate control over the development process. A Milstar satellite was lost when a typo in the software load tape was not detected during the development and testing. Control is also imposed by the management functions in an organization—the *Challenger* and *Columbia* losses, for example, involved inadequate controls in the launch-decision process.

While events reflect the *effects* of dysfunctional interactions and inadequate enforcement of safety constraints, the inadequate control itself is only indirectly reflected by the events—the events are the *result* of the inadequate control. The control structure itself must be examined to determine why it was inadequate to maintain the constraints on safe behavior and why the events occurred.

As an example, the unsafe behavior (hazard) in the *Challenger* loss was the release of hot propellant gases from the field joint. The miscreant O-ring was used to control the hazard—that is, its role was to seal a tiny gap in the field joint created by pressure at ignition. The loss occurred because the system design, including the O-ring, did not effectively impose the required constraint on the propellant gas release. Starting from here, there are then several questions that need to be answered to understand why the accident occurred and to obtain the information necessary to prevent future accidents. Why was this particular design unsuccessful in imposing the constraint, why was it chosen (what was the decision process), why was the flaw not found during development, and was there a different design that might have been more successful? These questions and others consider the original *design process*.

Understanding the accident also requires examining the contribution of the *operations process*. Why were management decisions made to launch despite warnings that it might not be safe to do so? One constraint that was violated during operations was the requirement to correctly handle feedback about any potential violation of the safety design constraints, in this case, feedback during operations that the control by the O-rings of the release of hot propellant gases from the field joints was not being adequately enforced by the design. There were several instances of feedback that were not adequately handled, such as data about O-ring blowby and erosion during previous shuttle launches and feedback by engineers who were concerned about the behavior of the O-rings in cold weather. Although the lack of redundancy provided by the second O-ring was known long before the loss of *Challenger*, that information was never incorporated into the NASA Marshall Space Flight Center database and was unknown by those making the launch decision. In addition, there was missing feedback about changes in the design and testing procedures during operations, such as the use of a new type of putty and the introduction of new O-ring leak checks without adequate verification that they satisfied system safety constraints on the field joints. As a final example, the control processes that ensured unresolved safety concerns were fully considered before each flight, that is, flight readiness reviews and other feedback channels to project management making flight decisions, were flawed.

Systems theory provides a much better foundation for safety engineering than the classic analytic reduction approach underlying event-based models of accidents. It provides a way forward to much more powerful and effective safety and risk analysis and management procedures that handle the inadequacies and needed extensions to current practice described in Chapter 2.

Combining a systems-theoretic approach to safety with system engineering processes will allow designing safety into the system as it is being developed or reengineered. System engineering provides an appropriate vehicle for this process because it rests on the same systems theory foundation and involves engineering the system as a whole.

3.5 Systems Engineering and Safety

The emerging theory of systems, along with many of the historical forces noted in Chapter 1, gave rise after World War II to a new emphasis in engineering, eventually called systems engineering. During and after the war, technology expanded rapidly and engineers were faced with designing and building more complex systems than had been attempted previously. Much of the impetus for the creation of this new discipline came from military programs in the 1950s and 1960s, particularly intercontinental ballistic missile (ICBM) systems. *Apollo* was the first nonmilitary government program in which systems engineering was recognized from the beginning as an essential function [23].

System Safety, as defined in MIL-STD-882, is a subdiscipline of system engineering. It was created at the same time and for the same reasons. The defense community tried using the standard safety engineering techniques on their complex new systems, but the limitations became clear when interface and component interaction problems went unnoticed until it was too late, resulting in many losses and near misses. When these early aerospace accidents were investigated, the causes of a large percentage of them were traced to deficiencies in design, operations, and management. Clearly, big changes were needed. System engineering along with its subdiscipline, System Safety, were developed to tackle these problems.

Systems theory provides the theoretical foundation for systems engineering, which views each system as an integrated whole even though it is composed of diverse, specialized components. The objective is to integrate the subsystems into the most effective system possible to achieve the overall objectives, given a prioritized set of design criteria. Optimizing the system design often requires making tradeoffs between these design criteria (goals).

The development of systems engineering as a discipline enabled the solution of enormously more complex and difficult technological problems than previously [136]. Many of the elements of systems engineering can be viewed merely as good engineering: It represents more a shift in emphasis than a change in content. In addition, while much of engineering is based on technology and science, systems engineering is equally concerned with overall management of the engineering process.

A systems engineering approach to safety starts with the basic assumption that some properties of systems, in this case safety, can only be treated adequately in the context of the social and technical system as a whole. A basic assumption of systems engineering is that optimization of individual components or subsystems will not in general lead to a system optimum; in fact, improvement of a particular subsystem may actually worsen the overall system performance because of complex, nonlinear interactions among the components. When each aircraft tries to optimize its path from its departure point to its destination, for example, the overall air transportation system throughput may not be optimized when they all arrive at a popular hub at the same time. One goal of the air traffic control system is to optimize the overall air transportation system throughput while, at the same time, trying to allow as much flexibility for the individual aircraft and airlines to achieve their goals. In the end, if system engineering is successful, everyone gains. Similarly, each pharmaceutical company acting to optimize its profits, which is a legitimate and reasonable company goal, will not necessarily optimize the larger societal *system* goal of producing safe and effective pharmaceutical and biological products to enhance public health. These system engineering principles are applicable even to systems beyond those traditionally thought of as in the engineering realm. The financial system and its meltdown starting in 2007 is an example of a social system that could benefit from system engineering concepts.

Another assumption of system engineering is that individual component behavior (including events or actions) cannot be understood without considering the components' role and interaction within the system as a whole. This basis for systems engineering has been stated as the principle that a system is more than the sum of its parts. Attempts to improve long-term safety in complex systems by analyzing and changing individual components have often proven to be unsuccessful over the long term. For example, Rasmussen notes that over many years of working in the field of nuclear power plant safety, he found that attempts to improve safety from models of local features were compensated for by people adapting to the change in an unpredicted way [166].

Approaches used to enhance safety in complex systems must take these basic systems engineering principles into account. Otherwise, our safety engineering approaches will be limited in the types of accidents and systems they can handle. At the same time, approaches that include them, such as those described in this book, have the potential to greatly improve our ability to engineer safer and more complex systems.

3.6 Building Safety into the System Design

System Safety, as practiced by the U.S. defense and aerospace communities as well as the new approach outlined in this book, fit naturally within the general systems engineering process and the problem-solving approach that a system view provides. This problem-solving process entails several steps. First, a need or problem is specified in terms of objectives that the system must satisfy along with criteria that can be used to rank alternative designs. For a system that has potential hazards, the objectives will include safety objectives and criteria along with high-level requirements and safety design constraints. The hazards for an automated train system, for example, might include the train doors closing while a passenger is in the doorway. The safety-related design constraint might be that obstructions in the path of a closing door must be detected and the door closing motion reversed.

After the high-level requirements and constraints on the system design are identified, a process of system synthesis takes place that results in a set of alternative designs. Each of these alternatives is analyzed and evaluated in terms of the stated objectives and design criteria, and one alternative is selected to be implemented. In practice, the process is highly iterative: The results from later stages are fed back to early stages to modify objectives, criteria, design alternatives, and so on. Of course, the process described here is highly simplified and idealized.

The following are some examples of basic systems engineering activities and the role of safety within them:

- *Needs analysis:* The starting point of any system design project is a perceived need. This need must first be established with enough confidence to justify the commitment of resources to satisfy it and understood well enough to allow appropriate solutions to be generated. Criteria must be established to provide a means to evaluate both the evolving and final system. If there are hazards associated with the operation of the system, safety should be included in the needs analysis.

- *Feasibility studies:* The goal of this step in the design process is to generate a set of realistic designs. This goal is accomplished by identifying the principal constraints and design criteria—including safety constraints and safety design criteria—for the specific problem being addressed and then generating plausible solutions to the problem that satisfy the requirements and constraints and are physically and economically feasible.

- *Trade studies:* In trade studies, the alternative feasible designs are evaluated with respect to the identified design criteria. A hazard might be controlled by any one of several safeguards: A trade study would determine the relative desirability of each safeguard with respect to effectiveness, cost, weight, size, safety and any other relevant criteria. For example, substitution of one material for another may reduce the risk of fire or explosion, but may also reduce reliability or efficiency. Each alternative design may have its own set of safety constraints (derived from the system hazards) as well as other performance goals and constraints that need to be assessed. Although, ideally, decisions should be based upon mathematical analysis, quantification of many of the key factors is often difficult, if not impossible, and subjective judgment often has to be used.

- *System architecture development and analysis:* In this step, the system engineers break down the system into a set of subsystems, together with the functions and constraints, including safety constraints, imposed upon the individual subsystem designs, the major system interfaces, and the subsystem interface topology. These aspects are analyzed with respect to desired system performance characteristics and constraints (again including safety constraints) and the process is iterated until an acceptable system design results. The preliminary design at the end of this process must be described in sufficient detail that subsystem implementation can proceed independently.

- *Interface analysis:* The interfaces define the functional boundaries of the system components. From a management standpoint, interfaces must (1) optimize visibility and control and (2) isolate components that can be implemented independently and for which authority and responsibility can be delegated [157]. From an engineering standpoint, interfaces must be designed to separate independent functions and to facilitate the integration, testing, and operation of the overall system. One important factor in designing the interfaces is safety, and safety analysis should be a part of the system interface analysis. Because interfaces tend to be particularly susceptible to design error and are implicated in the majority of accidents, a paramount goal of interface design is simplicity. Simplicity aids in ensuring that the interface can be adequately designed, analyzed, and tested prior to integration and that interface responsibilities can be clearly understood.

Any specific realization of this general systems engineering process depends on the engineering models used for the system components and the desired system qualities. For safety, the models commonly used to understand why and how accidents occur have been based on events, particularly failure events, and the use of reliability engineering techniques to prevent them. Part II of this book further details the alternative systems approach to safety introduced in this chapter while Part III provides techniques to perform many of these safety and system engineering activities.

Part II

STAMP: An Accident Model Based on Systems Theory

Part II introduces an expanded accident causality model based on the new assumptions in Chapter 2 and satisfying the goals stemming from them. The theoretical foundation for the new model is systems theory, as introduced in Chapter 3. Using this new causality model, called STAMP (Systems-Theoretic Accident Model and Processes), changes the emphasis in system safety from preventing failures to enforcing behavioral safety constraints. Component failure accidents are still included, but our conception of causality is extended to include component interaction accidents. Safety is reformulated as a control problem rather than a reliability problem. This change leads to much more powerful and effective ways to engineer safer systems, including the complex socio-technical systems of most concern today.

The three main concepts in this model—safety constraints, hierarchical control structures, and process models—are introduced first in Chapter 4. Then the STAMP causality model is described along with a classification of accident causes implied by the new model.

To provide additional understanding of STAMP, it is used to describe the causes of several very different types of losses—a friendly fire shootdown of a U.S. Army helicopter by a U.S. Air Force fighter jet over northern Iraq, the contamination of a public water system with E. coli bacteria in a small town in Canada, and the loss of a Milstar satellite. Chapter 5 presents the friendly fire accident analysis. The other accident analyses are contained in Appendices B and C.

Chapter 4

A Systems-Theoretic View of Causality

In the traditional causality models, accidents are considered to be caused by chains of failure events, each failure directly causing the next one in the chain. Part I explained why these simple models are no longer adequate for the more complex sociotechnical systems we are attempting to build today. The definition of accident causation needs to be expanded beyond failure events so that it includes component interaction accidents and indirect or systemic causal mechanisms.

The first step is to generalize the definition of an accident[1]. An *accident* is an unplanned and undesired loss event. That loss may involve human death and injury, but it may also involve other major losses including mission, equipment, financial, and information losses.

Losses result from component failures, disturbances external to the system, interactions among system components, and behavior of individual system components that lead to hazardous system states. Examples of hazards include the release of toxic chemicals from an oil refinery, a patient receiving a lethal dose of medicine, two aircraft violating minimum separation requirements, and commuter train doors opening between stations.[2]

In systems theory, emergent properties, such as safety, arise from the interactions among the system components. The emergent properties are controlled by imposing constraints on the behavior of and interactions among the components. Safety then becomes a *control* problem where the goal of the control is to enforce the safety constraints. Accidents result from inadequate control or enforcement of safety-related constraints on the development, design, and operation of the system.

At Bhopal, the safety constraint that was violated was that the MIC must not come in contact with water. In the Mars Polar Lander, the safety constraint was that the spacecraft must not impact the planet surface with more than a maximum force. In the batch chemical reactor accident described in Chapter 2, one safety constraint is a limitation on the temperature of the contents of the reactor.

The problem then becomes one of control where the goal is to control the behavior of the system by enforcing the safety constraints in its design and operation. Controls must be established to accomplish this goal. These controls need not necessarily involve a human or automated controller. Component behavior (including failures) and unsafe interactions may be controlled through physical

[1] A set of definitions used in this book can be found in Appendix A.

[2] Hazards are more carefully defined in Chapter 7

design, through process (such as manufacturing processes and procedures, maintenance processes, and operations), or through social controls. Social controls include organizational (management), governmental, and regulatory structures, but they may also be cultural, policy, or individual (such as self-interest) controls. As an example of the latter, one explanation that has been given for the 2009 financial crisis is that when investment banks went public, individual controls to reduce personal risk and long-term profits were eliminated and risk shifted to shareholders and others who had few and weak controls over those taking the risks.

In this framework, understanding why an accident occurred requires determining why the control was ineffective. Preventing future accidents requires shifting from a focus on preventing failures to the broader goal of designing and implementing controls that will enforce the necessary constraints.

The STAMP (System-Theoretic Accident Model and Processes) accident model is based on these principles. Three basic constructs underlie STAMP: safety constraints, hierarchical safety control structures, and process models.

4.1 Safety Constraints

The most basic concept in STAMP is not an event, but a constraint. Events leading to losses only occur because safety constraints were not successfully enforced.

The difficulty in identifying and enforcing safety constraints in design and operations has increased from the past. In many of our older and less automated systems, physical and operational constraints were often imposed by the limitations of technology and of the operational environments. Physical laws and the limits of our materials imposed natural constraints on the complexity of physical designs and allowed the use of passive controls.

In engineering, *passive controls* are those that maintain safety by their presence—basically the system design fails into a safe state or simple interlocks are used to limit the interactions among system components to safe ones. Some examples of passive controls that maintain safety by their presence are shields or barriers such as containment vessels, safety harnesses, hardhats, passive restraint systems in vehicles, and fences. Passive controls may also rely on physical principles, such as gravity, to fail into a safe state. An example is an old railway semaphore that used weights to ensure that if the cable (controlling the semaphore) broke, the arm would automatically drop into the STOP position. Other examples include mechanical relays designed to fail with their contacts open, and retractable landing gear for aircraft in which the wheels drop and lock in the landing position if the pressure system that raises and lowers them fails. For the batch chemical reactor example in Chapter 2, where the order valves are opened is crucial, designers might have used a physical interlock that allows the catalyst valve to be opened only if the water valve is open and allows the water valve to be closed only if the catalyst valve is closed.

In contract, *active controls* require some action(s) to provide protection: (1) *detection* of a hazardous event or condition (monitoring), (2) *measurement* of some variable(s),(3)interpretation of the measurement (*diagnosis*), and (4)*response* (recovery or fail-safe procedures), all of which must be completed before a loss occurs. These actions are usually implemented by a control system, which now commonly includes a computer.

Consider the simple passive safety control where the circuit for a high-power outlet is run through a door that shields the power outlet. When the door is opened, the circuit is broken and the power disabled. When the door is closed and the power enabled, humans cannot touch the high power outlet. Such a design is simple and fool proof. An active safety control design for the same high power source, requires some type of sensor to detect when the access door to the power outlet is opened and an active controller to issue a control command to cut the power. The failure modes for the active control system are greatly increased over the passive design, as is the complexity of the system component interactions. In the railway semaphore example, there must be a way to detect that the cable has broken (probably now a digital system is used instead of a cable so the failure of the digital signalling system must be detected) and some type of active controls used to warn operators to stop the train. The batch chemical reactor design described in Chapter 2 used a computer to control the valve opening and closing order instead of a simple mechanical interlock.

While simple examples are used here for practical reasons, the complexity of our designs is reaching and exceeding the limits of our intellectual manageability with a resulting increase in component interaction accidents and lack of enforcement of the system safety constraints. Even the relatively simple computer-based batch chemical reactor valve control design resulted in a component interaction accident. There are often very good reasons to use active controls instead of passive ones, including increased functionality, more flexibility in design, ability to operate over large distances, weight reduction, and so on. But the difficulty of the engineering problem is increased and more potential for design error is introduced.

A similar argument can be made for the interactions between operators and the processes they control. Cook [39] suggests that when controls were primarily mechanical and were operated by people located close to the operating process, proximity allowed sensory perception of the status of the process via direct physical feedback such as vibration, sound, and temperature (Figure 4.1). Displays were directly linked to the process and were essentially a physical extension of it. For example, the flicker of a gauge needle in the cab of a train indicated that (1) the engine valves were opening and closing in response to slight pressure fluctuations, (2) the gauge was connected to the engine, (3) the pointing indicator was free, and so on. In this way, the displays provided a rich source of information about the controlled process and the state of the displays themselves.

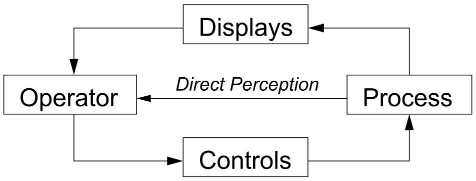


Figure 4.1: Operator has direct perception of process and mechanical controls

The introduction of electromechanical controls allowed operators to control processes from a greater distance (both physical and conceptual) than possible with pure mechanically linked controls (Figure 4.2). That distance, however, meant that operators lost a lot of direct information about the process—they could no longer sense the process state directly and the control and display surfaces no longer provided as rich a source of information about the process or the state of the controls themselves. The system designers had to synthesize and provide an image of the process state to the operators. An important new source of design errors was introduced by the need for the designers to determine beforehand what information the operator would need under all conditions to safely control the process. If the designers had not anticipated a particular situation could occur and provided for it in the original system design, they might also not anticipate the need of the operators for information about it during operations.

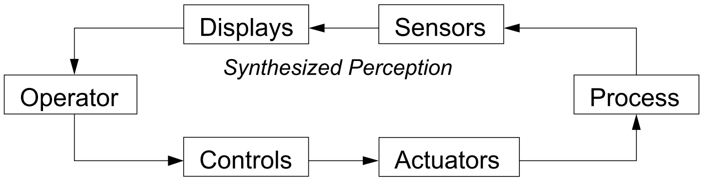


Figure 4.2: Operator has indirect information about process state and indirect controls

Designers also had to provide feedback on the actions of the operators and on any failures that might have occurred. The controls could now be operated without the desired effect on the process, and the operators might not know about it. Accidents started to occur due to incorrect feedback. For example, major accidents (including Three Mile Island) have involved the operators commanding a valve to open and receiving feedback that the valve had opened, when in reality it had not. In this case and others, the valves were wired to provide feedback that power had been applied to the valve, but not that the valve had actually opened. Not only could the design of the feedback about success and failures of control actions be misleading in these systems, but the return links were also subject to failure.

Electromechanical controls relaxed constraints on the system design allowing greater functionality (Figure 4.3). At the same time, they created new possibilities for designer and operator error that had not existed or were much less likely in mechanically controlled systems. The later introduction of computer and digital controls afforded additional advantages and removed even more constraints on the control system design—and introduced more possibility for error. Proximity in our old mechanical systems provided rich sources of feedback that involved almost all of the senses, enabling early detection of potential problems. We are finding it hard to capture and provide these same qualities in new systems that use automated controls and displays.

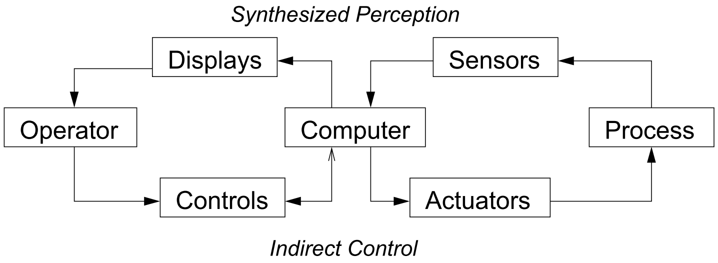


Figure 4.3: Operator has computer-generated displays and controls the process through a computer

It is the freedom from constraints that makes the design of such systems so difficult. Physical constraints enforced discipline and limited complexity in system design, construction, and modification. The physical constraints also shaped system design in ways that efficiently transmitted valuable physical component and process information to operators and supported their cognitive processes.

The same argument applies to the increasing complexity in organizational and social controls and in the interactions among the components of sociotechnical systems. Some engineering projects today employ thousands of engineers. The Joint Strike Fighter, for example, has 8000 engineers spread over most of the United States. Corporate operations have become global with greatly increased interdependencies and producing a large variety of products. A new holistic approach to safety, based on control and enforcing safety constraints in the entire sociotechnical system, is needed to ensure safety.

To accomplish this goal, system-level constraints must be identified and responsibility for enforcing them must be divided up and allocated to appropriate groups. For example, the members of one group might be responsible for performing hazard analyses. The manager of this group might be assigned responsibility for ensuring that the group has the resources, skills, and authority to perform such analyses and for ensuring that high-quality analyses result. Higher levels of management might have responsibility for budgets, for establishing corporate safety policies, and for providing oversight to ensure that safety policies and activities are being carried out successfully and that the information provided by the hazard analyses is used in design and operations.

During system and product design and development, the safety constraints will be broken down and sub-requirements or constraints allocated to the components of the design as it evolves. In the batch chemical reactor, for example, the system safety requirement is that the temperature in the reactor must always remain below a particular level. A design decision may be made to control this temperature using a reflux condenser. This decision leads to a new constraint that "Water must be flowing into the reflux condenser whenever catalyst is added to the reactor." After a decision is made about what component(s) will be responsible for operating the catalyst and water valves, additional requirements will be generated. If, for example, a decision is made to use software rather than (or in addition to) a physical interlock, the software must be assigned the responsibility for enforcing the constraint that "The software must ensure that the water valve is open whenever the catalyst valve is open."

In order to provide the level of safety demanded by society today, we first need to identify the safety constraints to enforce and then to design effective controls to enforce them. This process is much more difficult for today's complex and often high-tech systems than in the past and new techniques, such as those described in Part III, are going to be required to solve it, for example, methods to assist in generating the component safety constraints from the system safety constraints. The alternative—building only the simple electromechanical systems of the past or living with higher levels of risk—is for the most part not going to be considered an acceptable solution.

4.2 The Hierarchical Safety Control Structure

In systems theory (see Section 3.3), systems are viewed as hierarchical structures, where each level imposes constraints on the activity of the level beneath it—that is, constraints or lack of constraints at a higher level allow or control lower-level behavior.

Control processes operate between levels to control the processes at lower levels in the hierarchy. These control processes enforce the safety constraints for which the control process is responsible. Accidents occur when these processes provide inadequate control and the safety constraints are violated in the behavior of the lower-level components.

By describing accidents in terms of a hierarchy of control based on adaptive feedback mechanisms, adaptation plays a central role in the understanding and prevention of accidents.

At each level of the hierarchical structure, inadequate control may result from missing constraints (unassigned responsibility for safety), inadequate safety control commands, commands that were not executed correctly at a lower level, or inadequately communicated or processed feedback about constraint enforcement. For example, an operations manager may provide unsafe work instructions or procedures to the operators, or the manager may provide instructions that enforce the safety constraints but the operators may ignore them. The operations manager may not have the feedback channels established to determine that unsafe instructions were provided or that his or her safety-related instructions are not being followed.

Figure 4.4 shows a typical sociotechnical hierarchical safety control structure common in a regulated, safety-critical industry in the United States, such as air transportation. Each system, of course, must be modeled to include its specific features. Figure 4.4 has two basic hierarchical control structures—one for system development (on the left) and one for system operation (on the right)—with interactions between them. An aircraft manufacturer, for example, might only have system development under its immediate control, but safety involves both development and operational use of the aircraft, and neither can be accomplished successfully in isolation: Safety during operation depends partly on the original design and development and partly on effective control over operations. Communication channels may be needed between the two structures.[3] For example, aircraft manufacturers must communicate to their customers the assumptions about the operational environment upon which the safety analysis was based, as well as information about safe operating procedures. The operational environment (e.g., the commercial airline industry), in turn, provides feedback to the manufacturer about the performance of the system over its lifetime.

Between the hierarchical levels of each safety control structure, effective communication channels are needed, both a downward *reference channel* providing the information necessary to impose safety constraints on the level below and an upward *measuring channel* to provide feedback about how effectively the constraints are being satisfied (Figure 4.5). Feedback is critical in any open system in order to provide adaptive control. The controller uses the feedback to adapt future control commands to more readily achieve its goals.

Government, general industry groups, and the court system are the top two levels of each of the generic control structures shown in Figure 4.4. The government control structure in place to control development may differ from that controlling operations—responsibility for certifying the aircraft developed by aircraft manufacturers is assigned to one group at the FAA while responsibility for supervising airline operations is assigned to a different group. The appropriate constraints in each control structure and at each level will vary but in general may include technical design and process constraints, management constraints, manufacturing constraints, and operational constraints.

At the highest level in both the system development and system operation hierarchies are Congress and state legislatures.[4] Congress controls safety by passing laws and by establishing and funding government regulatory structures. Feedback as to the success of these controls or the need for additional ones comes in the form of government reports, congressional hearings and testimony, lobbying by various interest groups, and, of course, accidents.

The next level contains government regulatory agencies, industry associations, user associations, insurance companies, and the court system. Unions have always played an important role in

[3] Not all interactions between the two control structures are shown in the figure to simplify it and make it more readable.

[4] Obvious changes are required in the model for countries other than the United States. The United States is used in the example because of the author's familiarity with it.

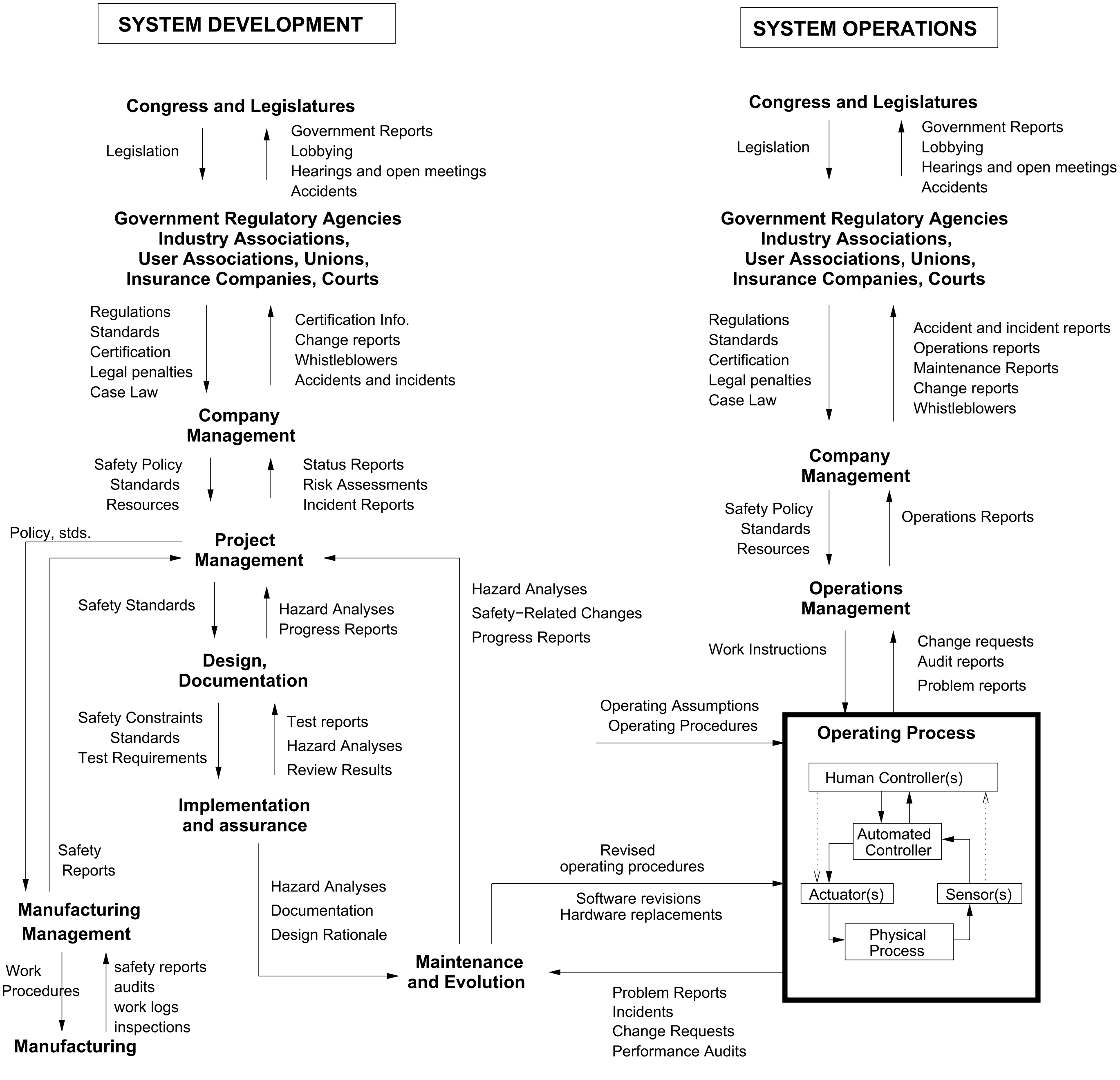


Figure 4.4: General Form of a Model of Socio-Technical Control

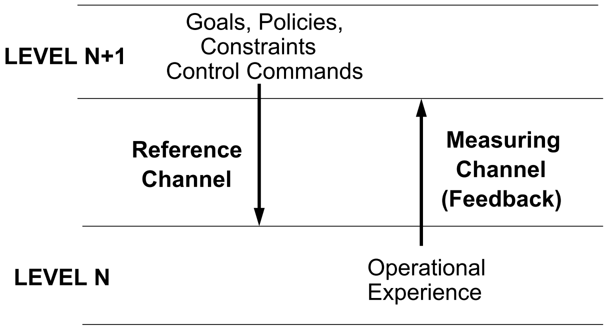

LEVEL N+1
Goals, Policies, Constraints Control Commands
Reference Channel
Measuring Channel (Feedback)
LEVEL N
Operational Experience

Figure 4.5: Communication Channels Between Control Levels.

ensuring safe operations, such as the air traffic controllers union in the air transportation system, or in ensuring worker safety in manufacturing. The legal system tends to be used when there is no regulatory authority and the public has no other means to encourage a desired level of concern for safety in company management. The constraints generated at this level and imposed on companies are usually in the form of policy, regulations, certification, standards (by trade or user associations), or threat of litigation. Where there is a union, safety-related constraints on operations or manufacturing may result from union demands and collective bargaining.

Company management takes the standards, regulations, and other general controls on its behavior and translates them into specific policy and standards for the company. Many companies have a general safety policy (it is required by law in Great Britain) as well as more detailed standards documents. Feedback may come in the form of status reports, risk assessments, and incident reports.

In the development control structure (shown on the left of Figure 4.4), company policies and standards are usually tailored and perhaps augmented by each engineering project to fit the needs of the particular project. The higher-level control process may provide only general goals and constraints and the lower levels may then add many details to operationalize the general goals and constraints given the immediate conditions and local goals. For example, while government or company standards may require a hazard analysis be performed, the system designers and documenters (including those designing the operational procedures and writing user manuals) may have control over the actual hazard analysis process used to identify specific safety constraints on the design and operation of the system. These detailed procedures may need to be approved by the level above.

The design constraints identified as necessary to control system hazards are passed to the implementers and assurers of the individual system components along with standards and other requirements. Success is determined through feedback provided by test reports, reviews, and various additional hazard analyses. At the end of the development process, the results of the hazard analyses as well as documentation of the safety-related design features and design rationale should be passed on to the maintenance group to be used in the system evolution and sustainment process.

A similar process involving layers of control is found in the system operation control structure. In addition, there will be (or at least should be) interactions between the two structures. For example, the safety design constraints used during development should form the basis for operating procedures and for performance and process auditing.

As in any control loop, time lags may affect the flow of control actions and feedback and may impact the effectiveness of the control loop in enforcing the safety constraints. For example, standards can take years to develop or change—a time scale that may keep them behind current technology and practice. At the physical level, new technology may be introduced in different parts of the system at different rates, which may result in *asynchronous evolution* of the control structure. In the accidental shootdown of two U.S. Army Black Hawk helicopters by two U.S. Air Force F-15s in the no-fly zone over northern Iraq in 1994, for example, the fighter jet aircraft and the helicopters were inhibited in communicating by radio because the F-15 pilots used newer jam-resistant radios that could not communicate with the older technology Army helicopter radios. Hazard analysis needs to include the influence of these time lags and potential changes over time.

A common way to deal with time lags leading to delays is to delegate responsibility to lower levels that are not subject to as great a delay in obtaining information or feedback from the measuring channels. In periods of quickly changing technology, time lags may make it necessary for the lower levels to augment the control processes passed down from above or to modify them to fit the current situation. Time lags at the lowest levels, as in the Black Hawk shootdown example, may require the use of feedforward control to overcome lack of feedback or may require temporary controls on behavior: Communication between the F-15s and the Black Hawks would have been possible if the F-15 pilots had been told to use an older radio technology available to them, as they were commanded to do for other types of friendly aircraft.

More generally, control structures always change over time, particularly those that include humans and organizational components. Physical devices also change with time, but usually much slower and in more predictable ways. If we are to handle social and human aspects of safety, then our accident causality models must include the concept of change. In addition, controls and assurance that the safety control structure remains effective in enforcing the constraints over time are required.

Control does not necessarily imply rigidity and authoritarian management styles. Rasmussen notes that control at each level may be enforced in a very prescriptive command and control structure or it may be loosely implemented as performance objectives with many degrees of freedom in how the objectives are met [166]. Recent trends from management by *oversight* to management by *insight* reflect differing levels of feedback control that are exerted over the lower levels and a change from prescriptive management control to management by objectives, where the objectives are interpreted and satisfied according to the local context.

Management insight, however, does not mean abdication of safety-related responsibility. In a Milstar satellite loss [152] and both the Mars Climate Orbiter [192] and Mars Polar Lander [94, 215] losses, the accident reports all note that a poor transition from oversight to insight was a factor in the losses. Attempts to delegate decisions and to manage by objectives require an explicit formulation of the value criteria to be used and an effective means for communicating the values down through society and organizations. In addition, the impact of specific decisions at each level on the objectives and values passed down need to be adequately and formally evaluated. Feedback is required to measure how successfully the functions are being performed.

Although regulatory agencies are included in the Figure 4.4 example, there is no implication that government regulation is required for safety. The only requirement is that responsibility for safety is distributed in an appropriate way throughout the sociotechnical system. In aircraft safety, for example, manufacturers play the major role while the FAA type certification authority simply provides oversight that safety is being successfully engineered into aircraft at the lower levels of the hierarchy. If companies or industries are unwilling or incapable of performing their public safety responsibilities, then government has to step in to achieve the overall public safety goals. But a much better solution is for company management to take responsibility as they have direct control over the system design and manufacturing and over operations.

The safety-control structure will differ among industries and examples are spread among the following chapters. Figure C.1 in Appendix C shows the control structure and safety constraints for the hierarchical water safety control system in Ontario, Canada. The structure is drawn on its side (as is more common for control diagrams) so that the top of the hierarchy is on the left side of the figure. The system hazard is exposure of the public to E. coli or other health-related contaminants through the public drinking water system; therefore, the goal of the safety control structure is to prevent such exposure. This goal leads to two system safety constraints:

1. Water quality must not be compromised.
2. Public health measures must reduce the risk of exposure if water quality is somehow compromised (such as notification and procedures to follow).

The physical processes being controlled by this control structure (shown at the right of the figure) are the water system, the wells used by the local public utilities, and public health. Details of the control structure are discussed in Appendix C, but appropriate responsibility, authority, and accountability must be assigned to each component with respect to the role it plays in the overall control structure. For example, the responsibility of the Canadian federal government is to establish a nationwide public health system and ensure it is operating effectively. The provincial government must establish regulatory bodies and codes, provide resources to the regulatory bodies, provide oversight and feedback loops to ensure that the regulators are doing their job adequately, and ensure that adequate risk assessment is conducted and effective risk management plans are in place. Local public utility operations must apply adequate doses of chlorine to kill bacteria, measure the chlorine residuals, and take further steps if evidence of bacterial contamination is found. While chlorine residuals are a quick way to get feedback about possible contamination, more accurate feedback is provided by analyzing water samples but takes longer (it has a greater time lag). Both have their uses in the overall safety control structure of the public water supply.

Safety control structures may be very complex: Abstracting and concentrating on parts of the overall structure may be useful in understanding and communicating about the controls. In examining different hazards, only subsets of the overall structure may be relevant and need to be considered in detail and the rest can be treated as the inputs to or the environment of the substructure. The only critical part is that the hazards must first be identified at the system level and the process must then proceed top-down and not bottom-up to identify the safety constraints for the parts of the overall control structure.

The operation of sociotechnical safety control structures at all levels is facing the stresses noted in Chapter 1, such as rapidly changing technology, competitive and time-to-market pressures, and changing public and regulatory views of responsibility for safety. These pressures can lead to a need for new procedures or new controls to ensure that required safety constraints are not ignored.

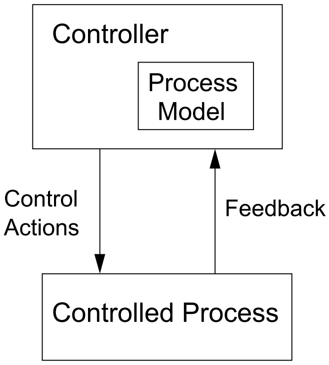


Figure 4.6: Every controller must contain a model of the process being controlled. Accidents can occur when the controller's process model does not match the state of the system being controlled and the controller issues unsafe commands.

4.3 Process Models

The third concept used in STAMP, along with safety constraints and hierarchical safety control structures, is process models. Process models are an important part of control theory. The four conditions required to control a process are described in Chapter 3. The first is a *goal*, which in STAMP is the safety constraints that must be enforced by each controller in the hierarchical safety control structure. The *action condition* is implemented in the (downward) control channels and the *observability condition* is embodied in the (upward) feedback or measuring channels. The final condition is the *model condition*: *Any* controller—human or automated—needs a model of the process being controlled to control it effectively (Figure 4.6).

At one extreme, this process model may contain only one or two variables, such as the model required for a simple thermostat, which contains the current temperature and the setpoint and perhaps a few control laws about how temperature is changed. At the other extreme, effective control may require a very complex model with a large number of state variables and transitions, such as the model needed to control air traffic.

Whether the model is embedded in the control logic of an automated controller or in the mental model maintained by a human controller, it must contain the same type of information: the required relationship among the system variables (the control laws), the current state (the current values of the system variables), and the ways the process can change state. This model is used to determine what control actions are needed, and it is updated through various forms of feedback. If the model of the room temperature shows that the ambient temperature is less than the setpoint, then the thermostat issues a control command to start a heating element. Temperature sensors provide feedback about the (hopefully rising) temperature. This feedback is used to update the thermostat's model of the current room temperature. When the setpoint is reached, the thermostat turns off the heating element. In the same way, human operators also require accurate process or mental models to provide safe control actions.

Component interaction accidents can usually be explained in terms of incorrect process models. For example, the Mars Polar Lander software thought the spacecraft had landed and issued a control instruction to shut down the descent engines. The captain of the Herald of Free Enterprise thought the ferry doors were closed and ordered the ship to leave the mooring. The pilots in the Cali Colombia B-757 crash thought R was the symbol denoting the radio beacon near Cali.

In general, accidents often occur, particularly component interaction accidents and accidents involving complex digital technology or human error, when the process model used by the controller (automated or human) does not match the actual state of the process and, as a result:

1. Unsafe control commands are given
2. Control actions required for safety are not provided
3. Potentially safe control commands are provided at the wrong time (too early or too late), or
4. Control is stopped too soon or applied too long.

These four types of inadequate control actions are used in the new hazard analysis technique described in Chapter 8.

A model of the process being controlled is required not just at the lower physical levels of the hierarchical control structure, but at all levels. In order to make proper decisions, the manager of an oil refinery may need to have a model of the current maintenance level of the safety equipment of the refinery, the state of safety training of the workforce, and the degree to which safety requirements are being followed or are effective, among other things. The CEO of the global oil conglomerate has a much less detailed model of the state of the refineries he controls but at the same time requires a broader view of the state of safety of all the corporate assets in order to make appropriate corporate-level decisions impacting safety.

Process models are not only used during operations but also during system development activities. Designers use both models of the system being designed and models of the development process itself. The developers may have an incorrect model of the system or software behavior necessary for safety or the physical laws controlling the system. Safety may also be impacted by developers' incorrect models of the development process itself.

As an example of the latter, a Titan/Centaur satellite launch system, along with the Milstar satellite it was transporting into orbit, was lost due to a typo in a load tape used by the computer to determine the attitude change instructions to issue to the engines. The information on the load tape was essentially part of the process model used by the attitude control software. The typo was not caught during the development process partly because of flaws in the developers' models of the testing process—each thought someone else was testing the software using the actual load tape when, in fact, nobody was (see Appendix B).

In summary, process models play an important role (1) in understanding why accidents occur and why humans provide inadequate control over safety-critical systems and (2) in designing safer systems.

4.4 STAMP

The STAMP (Systems-Theoretic Accident Model and Process) model of accident causation is built on these three basic concepts—safety constraints, a hierarchical safety control structure, and process models—along with basic systems theory concepts. All the pieces for a new causation model have been presented. It's now simply a matter of putting them together.

In STAMP, systems are viewed as interrelated components kept in a state of dynamic equilibrium by feedback control loops. Systems are not treated as static but as dynamic processes that are continually adapting to achieve their ends and to react to changes in themselves and their environment.

Safety is an emergent property of the system that is achieved when appropriate constraints on the behavior of the system and its components are satisfied. The original design of the system must not only enforce appropriate constraints on behavior to ensure safe operation, but the system must continue to enforce the safety constraints as changes and adaptations to the system design occur over time.

Accidents are the result of flawed processes involving interactions among people, societal and organizational structures, engineering activities, and physical system components that lead to violating the system safety constraints. The process leading up to an accident is described in STAMP in terms of an adaptive feedback function that fails to maintain safety as system performance changes over time to meet a complex set of goals and values.

Instead of defining safety management in terms of preventing component failures, it is defined as creating a safety control structure that will enforce the behavioral safety constraints and ensure its continued effectiveness as changes and adaptations occur over time. Effective safety (and risk) management may require limiting the types of changes that occur but the goal is to allow as much flexibility and performance enhancement as possible while still enforcing the safety constraints.

Accidents can be understood, using STAMP, by identifying the safety constraints that were violated and determining why the controls were inadequate in enforcing them. Understanding the Bhopal accident, for example, requires not simply determining why the maintenance personnel did not insert the slip blind, but why the controls that had been designed into the system to prevent the release of hazardous chemicals and to mitigate the consequences of such occurrences—including maintenance procedures and oversight of maintenance processes, refrigeration units, gauges and other monitoring units, a vent scrubber, water spouts, a flare tower, safety audits, alarms and practice alerts, emergency procedures and equipment, and others—were not successful.

STAMP not only allows consideration of more accident causes than simple component failures, but it also allows more sophisticated analysis of failures and component failure accidents. Component failures may result from inadequate constraints on the manufacturing process; inadequate engineering design such as missing or incorrectly implemented fault tolerance; lack of correspondence between individual component capacity (including human capacity) and task requirements; unhandled environmental disturbances (e.g., electromagnetic interference or EMI); inadequate maintenance; physical degradation (wearout); and so on.

Component failures may be prevented by increasing the integrity or resistance of the component to internal or external influences or by building in safety margins or safety factors. They may also be avoided by operational controls, such as operating the component within its design envelope and by periodic inspections and preventive maintenance. Manufacturing controls can reduce deficiencies or flaws introduced during the manufacturing process. The effects of physical component failure on system behavior may be eliminated or reduced by using redundancy. The important difference from other causality models is that STAMP goes beyond simply blaming component failure for accidents by requiring that the reasons be identified for why those failures occurred (including systemic factors) and led to an accident, that is, why the controls instituted for preventing such failures or for minimizing their impact on safety) were missing or inadequate. And it includes other types of accident causes, such as component interaction accidents, which are becoming more frequent with the introduction of new technology and new roles for humans in system control.

STAMP does not lend itself to a simple graphic representation of accident causality (see 4.7).

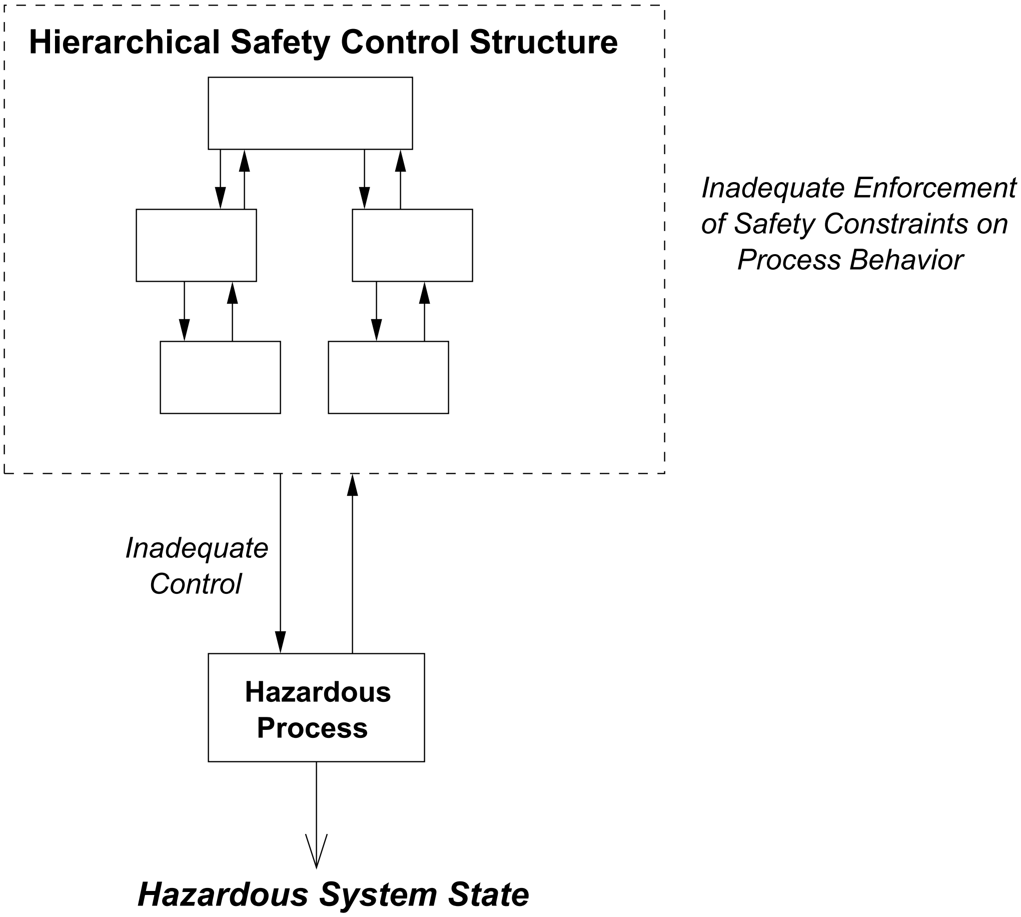

Hierarchical Safety Control Structure
Inadequate Enforcement of Safety Constraints on Process Behavior
Inadequate Control
Hazardous Process
Hazardous System State

Figure 4.7: Accidents result from inadequate enforcement of the behavioral safety constraints on the process

While dominoes, event chains, and holes in Swiss cheese are very compelling because they are easy to grasp, they oversimplify causality and thus the approaches used to prevent accidents.

4.5 A General Classification of Accident Causes

Starting from the basic definitions in STAMP, the general causes of accidents can be identified using basic systems and control theory. The resulting classification is useful in accident analysis and accident prevention activities.

Accidents in STAMP are the result of a complex process that results in the system behavior violating the safety constraints. The safety constraints are enforced by the control loops between the various levels of the hierarchical control structure that are in place during design, development, manufacturing, and operations.

Using the STAMP causality model, if there is an accident, one or more of the following must have occurred:

1. The safety constraints were not enforced by the controller.
   a. The control actions necessary to enforce the associated safety constraint at each level of the sociotechnical control structure for the system were not provided,
   b. The necessary control actions were provided but at the wrong time (too early or too late), stopped too soon, or applied too long, or
   c. Unsafe control actions were provided that caused a violation of the safety constraints,
2. Appropriate control actions were provided but not followed.

These same general factors apply at each level of the sociotechnical control structure, but the interpretation (application) of the factor at each level may differ.

Classification of accident causal factors starts by examining each of the basic components of a control loop (refer to Figure 3.2) and determining how their improper operation may contribute to the general types of inadequate control.

Figure 4.8 shows the classification. The causal factors in accidents can be divided into three general categories: (1) the controller operation, (2) the behavior of actuators and controlled processes, and (3) communication and coordination among controllers and decision makers. When humans are involved in the control structure, context and behavior-shaping mechanisms also play an important role in causality.

4.5.1 Controller Operation

Controller operation has three primary parts: control inputs and other relevant external information sources, the control algorithms, and the process model. Inadequate, ineffective, or missing control actions necessary to enforce the safety constraints and ensure safety can stem from flaws in each of these parts. For human controllers and actuators, context is also an important factor.

Unsafe Inputs (①in Figure 4.8): Each controller in the hierarchical control structure is itself controlled by higher-level controllers. The control actions and other information provided by the higher level and required for safe behavior may be missing or wrong. Using the Black Hawk friendly fire example again, the F-15 pilots patrolling the no-fly zone were given instructions to switch to a non-jammed radio mode for a list of aircraft types that did not have the ability to interpret jammed broadcasts. Black Hawk helicopters had not been upgraded with new anti-jamming technology but were omitted from the list and so could not hear the F-15 radio broadcasts. Other types of missing or wrong non-control inputs may also affect the operation of the controller.

Unsafe Control Algorithms (② in Figure 4.8): Algorithms in this sense are both the procedures designed by engineers for hardware controllers and the procedures that human controllers use. Control algorithms may not enforce safety constraints because the algorithms are inadequately designed originally, the process may change and the algorithms become unsafe, or the control algorithms may be inadequately modified by maintainers if the algorithms are automated or through various types of natural adaptation if they are implemented by humans. Human control algorithms are affected by initial training, by the procedures provided to the operators to follow, and by feedback and experimentation over time (see Figure 2.9).

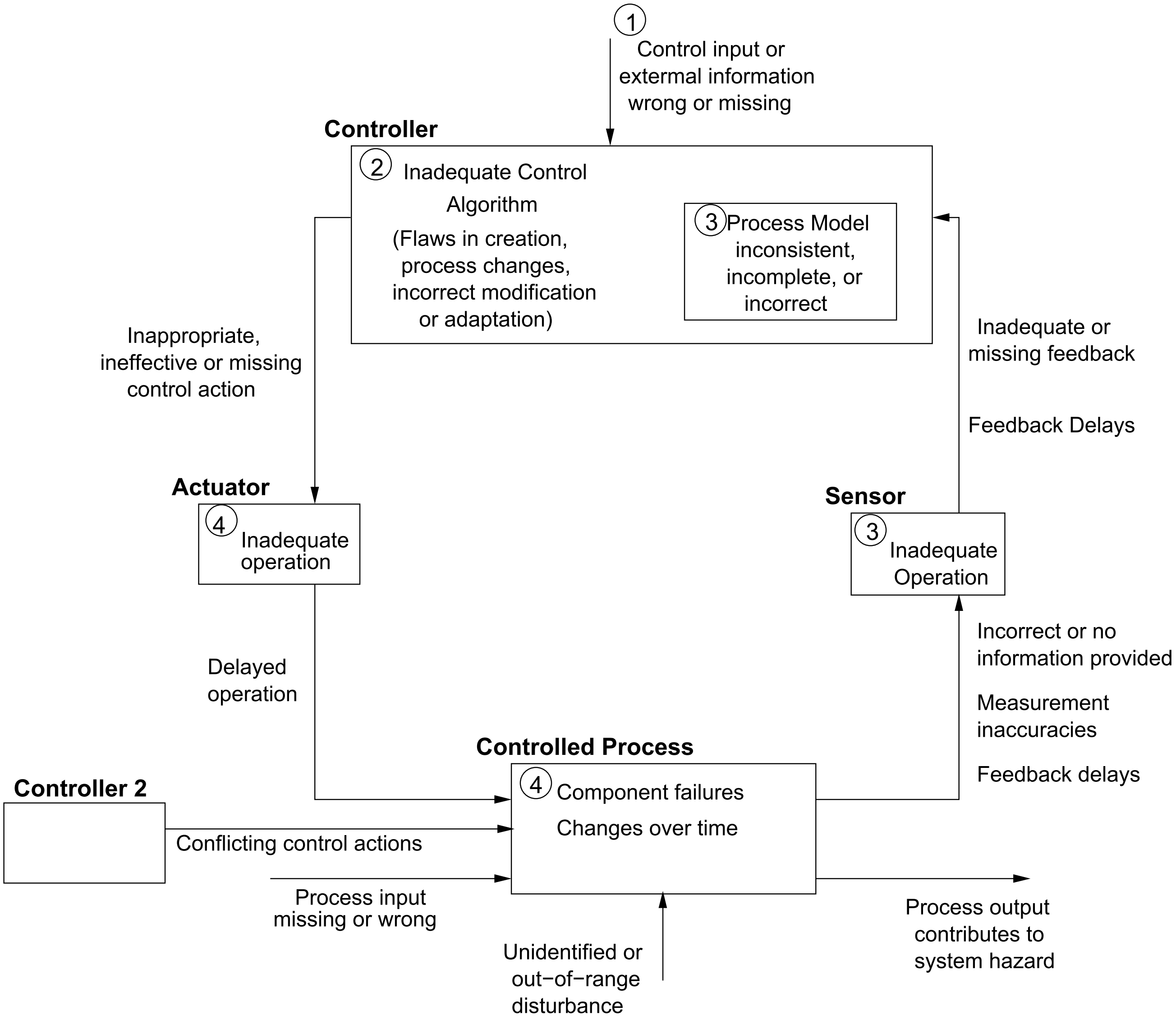


Figure 4.8: A Classification of Control Flaws Leading to Hazards

Time delays are an important consideration in designing control algorithms. Any control loop includes time lags, such as the time between the measurement of process parameters and receiving those measurements or between issuing a command and the time the process state actually changes. For example, pilot response delays are important time lags that must be considered in designing the control function for TCAS[5] or other aircraft systems, as are time lags in the controlled process—the aircraft trajectory, for example—caused by aircraft performance limitations.

Delays may not be directly observable, but may need to be inferred. Depending on where in the feedback loop the delay occurs, different control algorithms are required to cope with the delays [24]: dead time and time constants require an algorithm that makes it possible to predict when an action is needed before the need. Feedback delays generate requirements to predict when a prior control action has taken effect and when resources will be available again. Such requirements may impose the need for some type of open loop or feedforward strategy to cope with delays. When time delays are not adequately considered in the control algorithm, accidents can result.

Leplat has noted that many accidents relate to *asynchronous evolution* [111], where one part of a system (in this case the hierarchical safety control structure) changes without the related necessary changes in other parts. Changes to subsystems may be carefully designed, but consideration of their effects on other parts of the system, including the safety control aspects, may be neglected or inadequate. Asynchronous evolution may also occur when one part of a properly designed system deteriorates.

In both these cases, the erroneous expectations of users or system components about the behavior of the changed or degraded subsystem may lead to accidents. The Ariane 5 trajectory changed from that of the Ariane 4, but the inertial reference system software was not changed. As a result, an assumption of the inertial reference software was violated and the spacecraft was lost shortly after launch. One factor in the loss of contact with SOHO (SOlar Heliospheric Observatory), a scientific spacecraft, in 1998 was the failure to communicate to operators that a functional change had been made in a procedure to perform gyro spin down. The Black Hawk friendly fire accident (analyzed in Chapter 5) had several examples of asynchronous evolution, for example the mission changed and an individual key to communication between the Air Force and Army left: leaving the safety control structure without an important component.

Communication is a critical factor here as well as monitoring for changes that may occur and feeding back this information to the higher-level control. For example, the safety analysis process that generates constraints always involves some basic assumptions about the operating environment of the process. When the environment changes such that those assumptions are no longer true, as in the Ariane 5 and SOHO examples above, the controls in place may become inadequate. Embedded pacemakers provide another example. These devices were originally assumed to be used only in adults, who would lie quietly in the doctor's office while the pacemaker was being "programmed." Later these devices began to be used in children, and the assumptions under which the hazard analysis was conducted and the controls were designed no longer held and needed to be revisited. A requirement for effective updating of the control algorithms is that the assumptions of the original (and subsequent) analysis are recorded and retrievable.

Inconsistent, Incomplete, or Incorrect Process Models (③ in Figure 4.8): Section 4.3 stated that effective control is based on a model of the process state. Accidents, particularly

[5] TCAS (Traffic alert and Collision Avoidance System) is an airborne system used to avoid collisions between aircraft. More details about TCAS can be found in Chapter 10.

component interaction accidents, most often result from inconsistencies between the models of the process used by the controllers (both human and automated) and the actual process state. When the controller's model of the process (either the human mental model or the software or hardware model) diverges from the process state, erroneous control commands (based on the incorrect model) can lead to an accident: for example, (1) the software does not know that the plane is on the ground and raises the landing gear or (2) the controller (automated or human) does not identify an object as friendly and shoots a missile at it or (3) the pilot thinks the aircraft controls are in *speed* mode but the computer has changed the mode to *open descent* and the pilot behaves inappropriately for that mode or (4) the computer does not think the aircraft has landed and overrides the pilots' attempts to operate the braking system. All of these examples have actually occurred.

The mental models of the system developers are also important. During software development, for example, the programmers' models of required behavior may not match engineers' models (commonly referred to as a software requirements error), or the software may be executed on computer hardware or may control physical systems during operations that differs from what was assumed by the programmer and used during testing. The situation becomes more even complicated when there are multiple controllers (both human and automated) because each of their process models must also be kept consistent.

The most common form of inconsistency occurs when one or more of the process models is incomplete in terms of not defining appropriate behavior for all possible process states or all possible disturbances, including unhandled or incorrectly handled component failures. Of course, no models are complete in the absolute sense: The goal is to make them complete enough that no safety constraints are violated when they are used. Criteria for completeness in this sense are presented in *Safeware*, and completeness analysis is integrated into the new hazard analysis method as described in Chapter 9.

How does the process model become inconsistent with the actual process state? The process model designed into the system (or provided by training if the controller is human) may be wrong from the beginning, there may be missing or incorrect feedback for updating the process model as the controlled process changes state, the process model may be updated incorrectly (an error in the algorithm of the controller), or time lags may not be accounted for. The result can be uncontrolled disturbances, unhandled process states, inadvertent commanding of the system into a hazardous state, unhandled or incorrectly handled controlled process component failures, and so on.

Feedback is critically important to the safe operation of the controller. A basic principle of system theory is that no control system will perform better than its measuring channel. Feedback may be missing or inadequate because such feedback is not included in the system design, flaws exist in the monitoring or feedback communication channel, the feedback is not timely, or the measuring instrument operates inadequately.

A contributing factor cited in the Cali B-757 accident report, for example, was the omission of the waypoints[6] behind the aircraft from cockpit displays, which contributed to the crew not realizing that the waypoint for which they were searching was behind them (missing feedback). The model of the Ariane 501 attitude used by the attitude control software became inconsistent with the launcher attitude when an error message sent by the inertial reference system was interpreted by the attitude control system as data (incorrect processing of feedback), causing the spacecraft onboard computer to issue an incorrect and unsafe command to the booster and main engine nozzles.

[6] A *waypoint* is a set of coordinates that identify a point in physical space.

Other reasons for the process models to diverge from the true system state may be more subtle. Information about the process state has to be inferred from measurements. For example, in the TCAS II aircraft collision avoidance system, relative range positions of other aircraft are computed based on round-trip message propagation time. The theoretical control function (control law) uses the true values of the controlled variables or component states (e.g., true aircraft positions). However, at any time, the controller has only measured values, which may be subject to time lags or inaccuracies. The controller must use these measured values to infer the true conditions in the process and, if necessary, to derive corrective actions to maintain the required process state. In the TCAS example, sensors include on-board devices such as altimeters that provide measured altitude (not necessarily true altitude) and antennas for communicating with other aircraft. The primary TCAS actuator is the pilot, who may or may not respond to system advisories. The mapping between the measured or assumed values and the true values can be flawed.

To summarize, process models can be incorrect from the beginning—where correct is defined in terms of consistency with the current process state and with the models being used by other controllers—or they can become incorrect due to erroneous or missing feedback or measurement inaccuracies. They may also be incorrect only for short periods of time due to time lags in the process loop.

4.5.2 Actuators and Controlled Processes (④in Figure 4.8)

The factors discussed so far have involved inadequate control. The other case occurs when the control commands maintain the safety constraints, but the controlled process may not implement these commands. One reason might be a failure or flaw in the reference channel, that is, in the transmission of control commands. Another reason might be an actuator or controlled component fault or failure. A third is that the safety of the controlled process may depend on inputs from other system components, such as power, for the execution of the control actions provided. If these process inputs are missing or inadequate in some way, the controller process may be unable to execute the control commands and accidents may result. Finally, there may be external disturbances that are not handled by the controller.

In a hierarchical control structure, the actuators and controlled process may themselves be a controller of a lower-level process. In this case, the flaws in executing the control are the same described above for a controller.

Once again, these types of flaws do not simply apply to operations or to the technical system but also to system design and development. For example, a common flaw in system development is that the safety information gathered or created by the system safety engineers (the hazards and the necessary design constraints to control them) is inadequately communicated to the system designers and testers or that flaws exist in the use of this information in the system development process.

4.5.3 Coordination and Communication Among Controllers and Decision Makers

When there are multiple controllers (human and/or automated), control actions may be inadequately coordinated, including unexpected side effects of decisions or actions or conflicting control actions. Communication flaws play an important role here.

Leplat suggests that accidents are most likely in *overlap areas* or in *boundary areas* or where two or more controllers (human or automated) control the same process or processes with common boundaries (Figure 4.9) [111]. In both boundary and overlap areas, the potential exists for ambiguity and for conflicts among independent decisions.

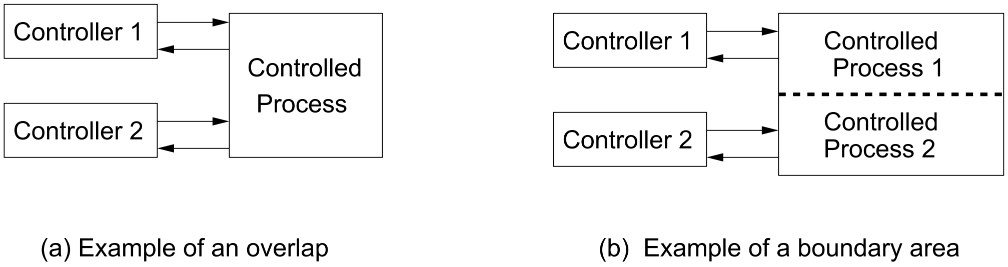


Figure 4.9: Problems often occur when there is shared control over the same process or at the boundary areas between separately controlled processes.

Responsibility for the control functions in boundary areas is often poorly defined. For example, Leplat cites an iron and steel plant where frequent accidents occurred at the boundary of the blast furnace department and the transport department. One conflict arose when a signal informing transport workers of the state of the blast furnace did not work and was not repaired because each department was waiting for the other to fix it. Faverge suggests that such dysfunction can be related to the number of management levels separating the workers in the departments from a common manager: The greater the distance, the more difficult the communication, and thus the greater the uncertainty and risk.

Coordination problems in the control of boundary areas are rife. As mentioned earlier, a Milstar satellite was lost due to inadequate attitude control of the Titan/Centaur launch vehicle, which used an incorrect process model based on erroneous inputs on a software load tape. After the accident, it was discovered that nobody had tested the software using the actual load tape—each group involved in testing and assurance had assumed some other group was doing so. In the system development process, system engineering and mission assurance activities were missing or ineffective, and a common control or management function was quite distant from the individual development and assurance groups (see Chapter B). One factor in the loss of the Black Hawk helicopters to friendly fire over northern Iraq was that the helicopters normally flew only in the boundary areas of the no-fly zone and procedures for handling aircraft in those areas were ill defined. Another factor was that an Army base controlled the flights of the Black Hawks while an Air Force base controlled all the other components of the airspace. A common control point once again was high above where the accident occurred in the control structure. In addition, communication problems existed between the Army and Air Force bases at the intermediate control levels.

Overlap areas exist when a function is achieved by the cooperation of two controllers or when two controllers exert influence on the same object. Such overlap creates the potential for conflicting control actions (dysfunctional interactions among control actions). Leplat cites a study of the steel industry that found 67 percent of technical incidents with material damage occurred in areas of co-activity, although these represented only a small percentage of the total activity areas. In an A320 accident in Bangalore, India, the pilot had disconnected his flight director during approach and assumed that the co-pilot would do the same. The result would have been a mode configuration in which airspeed is automatically controlled by the autothrottle (the *speed* mode), which is the recommended procedure for the approach phase. However, the co-pilot had not turned off his flight director, which meant that *open descent* mode became active when a lower altitude was selected instead of *speed* mode, eventually contributing to the crash of the aircraft short of the runway [180]. In the Black Hawks' shootdown by friendly fire, the aircraft surveillance officer (ASO) thought she was responsible only for identifying and tracking aircraft south of the 36th Parallel while the air traffic controller for the area north of the 36th Parallel thought the ASO was also tracking and identifying aircraft in his area and acted accordingly.

In 2002, two aircraft collided over southern Germany. An important factor in the accident was the lack of coordination between the airborne TCAS (collision avoidance) system and the ground air traffic controller. They each gave different and conflicting advisories on how to avoid a collision. If both pilots had followed one or the other, the loss would have been avoided, but one followed the TCAS advisory and the other followed the ground air traffic control advisory.

4.5.4 Context and Environment

Flawed human decision making can result from incorrect information and inaccurate process models as described above. But human behavior is also greatly impacted by the context and environment in which the human is working. These factors have been called "behavior shaping mechanisms." While value systems and other influences on decision making can be considered to be inputs to the controller, describing them in this way oversimplifies their role and origin. A classification of the contextual and behavior-shaping mechanisms is premature at this point, but relevant principles and heuristics are elucidated throughout the rest of the book.

4.6 Applying the New Model

To summarize, STAMP focuses particular attention on the role of constraints in safety management. Accidents are seen as resulting from inadequate control or enforcement of constraints on safety-related behavior at each level of the system development and system operations control structures. Accidents can be understood in terms of why the controls that were in place did not prevent or detect maladaptive changes.

Accident causal analysis based on STAMP starts with identifying the safety constraints that were violated and then determines why the controls designed to enforce the safety constraints were inadequate or, if they were potentially adequate, why the system was unable to exert appropriate control over their enforcement.

In this conception of safety, there is no "root cause." Instead, the accident "cause" consists of an inadequate safety control structure that under some circumstances leads to the violation of a behavioral safety constraint. Preventing future accidents requires re-engineering or designing the safety control structure to be more effective.

Because the safety control structure and the behavior of the individuals in it, like any physical or social system, changes over time, accidents must be viewed as dynamic processes. Looking only at the time of the proximal loss events distorts and omits from view the most important aspects of the larger accident process that are needed to prevent reoccurrences of losses from the same causes in the future. Without that view, we only see and fix the symptoms, that is, the results of the flawed processes and inadequate safety control structure without getting to the sources of those symptoms.

To understand the dynamic aspects of accidents, the process leading to the loss can be viewed as an adaptive feedback function where the safety control system performance degrades over time as the system attempts to meet a complex set of goals and values. Adaptation is critical in understanding accidents, and the adaptive feedback mechanism inherent in the model allows a STAMP analysis to incorporate adaptation as a fundamental system property.

We have found in practice that using this model helps us to separate factual data from the interpretations of that data: While the events and physical data involved in accidents may be clear, their importance and the explanations for why the factors were present are often subjective as is the selection of the events to consider.

STAMP models are also more complete than most accident reports and other models, for example see [9, 88, 141]. Each of the explanations for the incorrect FMS input of *R* in the Cali American Airlines accident described in Chapter 2, for example, appears in the STAMP analysis of that accident at the appropriate levels of the control structure where they operated. The use of STAMP not only helps to identify the factors but also to understand the relationships among them.

While STAMP models will probably not be useful in lawsuits as they do not assign blame for the accident to a specific person or group, they do provide more help in understanding accidents by forcing examination of each part of the sociotechnical system to see how it contributed to the loss—and there will usually be contributions at each level. Such understanding should help in learning how to engineer safer systems, including the technical, managerial, organizational, and regulatory aspects.

To accomplish this goal, a framework for classifying the factors that lead to accidents was derived from the basic underlying conceptual accident model (see Figure 4.8). This classification can be used in identifying the factors involved in a particular accident and in understanding their role in the process leading to the loss. The accident investigation after the Black Hawk shootdown (analyzed in detail in the next chapter) identified 130 different factors involved in the accident. In the end, only the AWACS senior director was court-martialed, and he was acquitted. The more one knows about an accident process, the more difficult it is to find one person or part of the system responsible, but the easier it is to find effective ways to prevent similar occurrences in the future.

STAMP is useful not only in analyzing accidents that have occurred but in developing new and potentially more effective system engineering methodologies to prevent accidents. Hazard analysis can be thought of as investigating an accident before it occurs. Traditional hazard analysis techniques, such as fault tree analysis and various types of failure analysis techniques, do not work well for very complex systems, for software errors, human errors, and system design errors. Nor do they usually include organizational and management flaws. The problem is that these hazard analysis techniques are limited by a focus on failure events and the role of component failures in accidents; they do not account for component interaction accidents, the complex roles that software and humans are assuming in high-tech systems, the organizational factors in accidents, and the indirect relationships between events and actions required to understand why accidents occur.

STAMP provides a direction to take in creating these new hazard analysis and prevention techniques. Because in a system accident model everything starts from constraints, the new approach focuses on identifying the constraints required to maintain safety; identifying the flaws in the control structure that can lead to an accident (inadequate enforcement of the safety constraints); and then designing a control structure, physical system and operating conditions that enforces the constraints.

Such hazard analysis techniques augment the typical failure-based design focus and encourage a wider variety of risk reduction measures than simply adding redundancy and overdesign to deal with component failures. The new techniques also provide a way to implement *safety-guided design* so that safety analysis guides the design generation rather than waiting until a design is complete to discover it is unsafe. Part III describes ways to use techniques based on STAMP to prevent accidents through system design, including design of the operating conditions and the safety management control structure.

STAMP can also be used to improve performance analysis. Performance monitoring of complex systems has created some dilemmas. Computers allow the collection of massive amounts of data, but analyzing that data to determine whether the system is moving toward the boundaries of safe behavior is difficult. The use of an accident model based on system theory and the basic concept of safety constraints may provide directions for identifying appropriate safety metrics and leading indicators; determining whether control over the safety constraints is adequate; evaluating the assumptions about the technical failures and potential design errors, organizational structure, and human behavior underlying the hazard analysis; detecting errors in the operational and environmental assumptions underlying the design and the organizational culture; and identifying any maladaptive changes over time that could increase risk of accidents to unacceptable levels.

Finally, STAMP points the way to very different approaches to risk assessment. Currently, risk assessment is firmly rooted in the probabilistic analysis of failure events. Attempts to extend current PRA techniques to software and other new technology, to management, and to cognitively complex human control activities have been disappointing. This way forward may lead to a dead end. Significant progress in risk assessment for complex systems will require innovative approaches starting from a completely different theoretical foundation.

Chapter 5

A Friendly Fire Accident

The goal of STAMP is to assist in understanding why accidents occur and to use that understanding to create new and better ways to prevent losses. This chapter and several of the appendices provide examples of how STAMP can be used to analyze and understand accident causation. The particular examples were selected to demonstrate the applicability of STAMP to very different types of systems and industries.

This chapter delves into the causation of the loss of a U.S. Army Black Hawk helicopter and all its occupants from friendly fire by a U.S. Air Force F-15 over northern Iraq in 1994. This example was chosen because the controversy and multiple viewpoints and books about the shootdown provide the information necessary to create most of the STAMP analysis. Accident reports often leave out important causal information (as did the official accident report in this case). Because of the nature of the accident, most of the focus is on operations. Appendix B presents an example of an accident where engineering development plays an important role. Social issues involving public health are the focus of the accident analysis in Appendix C.

5.1 Background

After the Persian Gulf War, Operation Provide Comfort (OPC) was created as a multinational humanitarian effort to relieve the suffering of hundreds of thousands of Kurdish refugees who fled into the hills of northern Iraq during the war. The goal of the military efforts was to provide a safe haven for the resettlement of the refugees and to ensure the security of relief workers assisting them. The formal mission statement for OPC read: "To deter Iraqi behavior that may upset peace and order in northern Iraq."

In addition to operations on the ground, a major component of OPC's mission was to occupy the airspace over northern Iraq. To accomplish this task, a no-fly-zone (also called the TAOR or Tactical Area of Responsibility) was established that included all airspace within Iraq north of the 36th Parallel (see Figure 5.1). Air operations were led by the Air Force to prohibit Iraqi aircraft from entering the no-fly zone while ground operations were organized by the Army to provide humanitarian assistance to the Kurds and other ethnic groups in the area.

U.S., Turkish, British, and French fighter and support aircraft patrolled the no-fly zone daily to prevent Iraqi warplanes from threatening the relief efforts. The mission of the Army helicopters was to support the ground efforts; the Army used them primarily for troop movement, resupply,

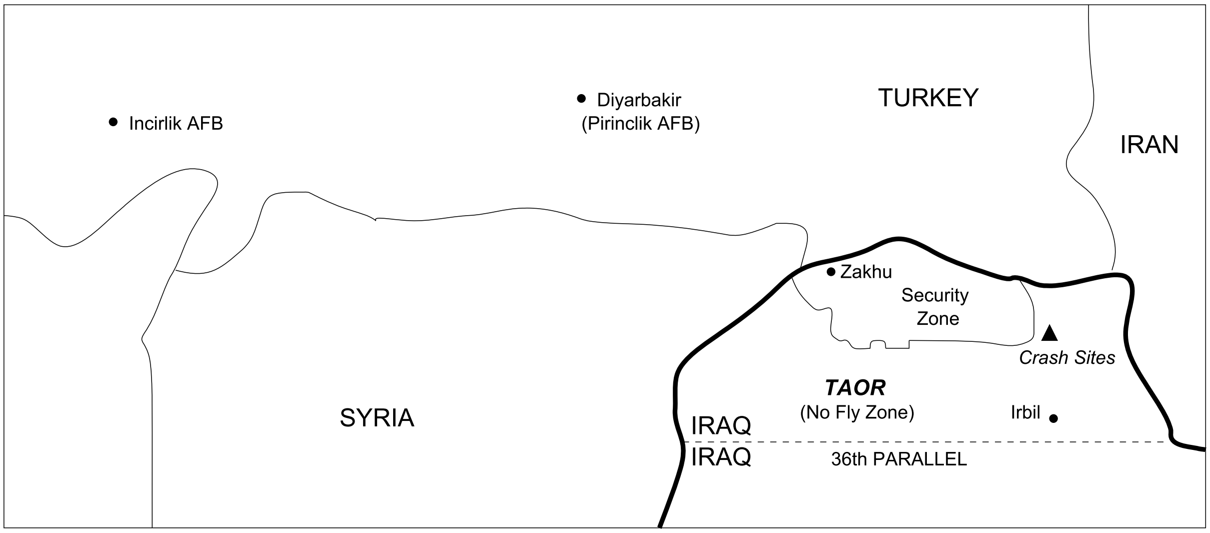


Figure 5.1: The No-Fly Z-one and Relevant Surrounding Locations

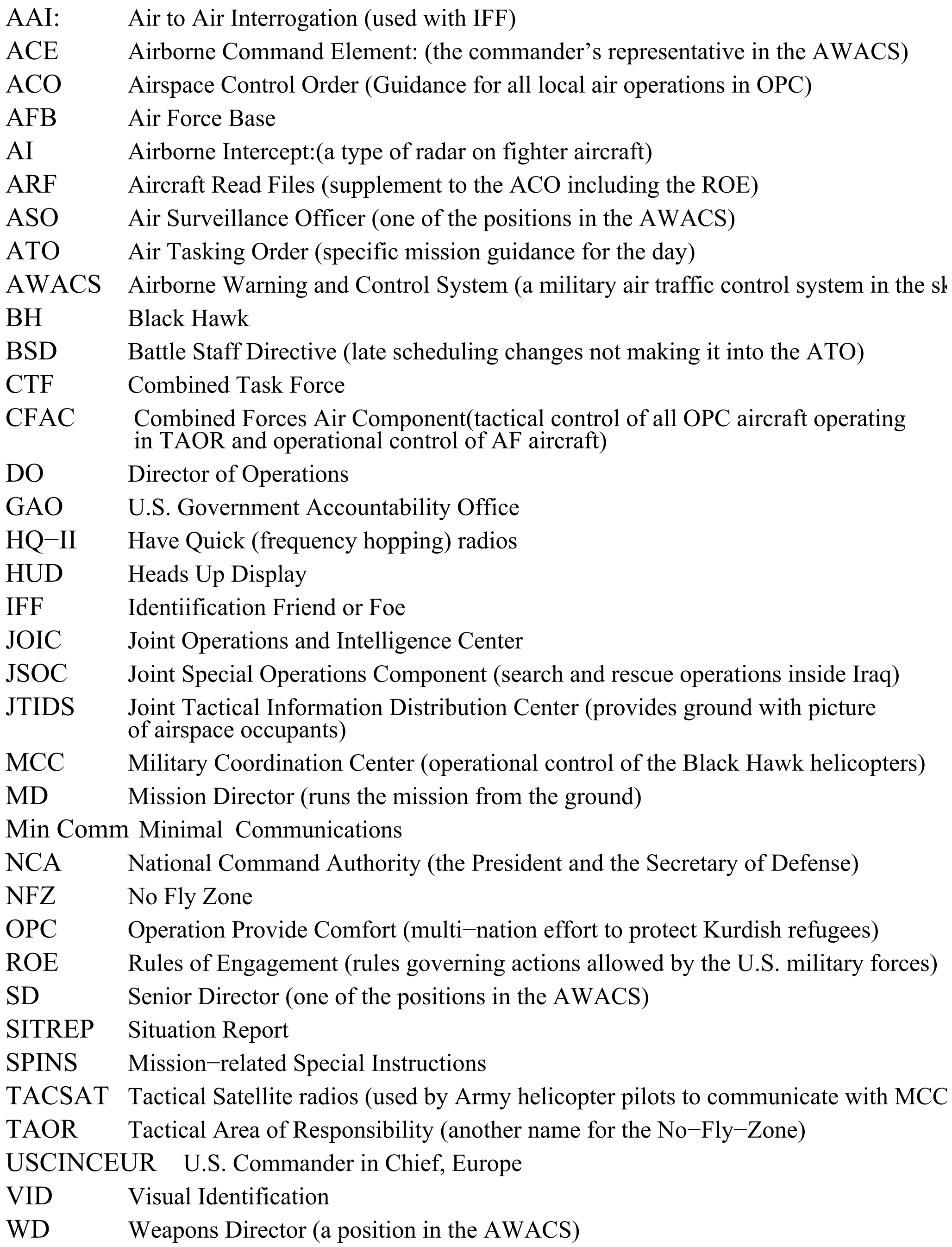
and medical evacuation.

On April 15, 1994, after nearly three years of daily operations over the TAOR (Tactical Area of Responsibility), two U.S. Air Force F-15s patrolling the area shot down two U.S. Army Black Hawk helicopters, mistaking them for Iraqi Hind helicopters. The Black Hawks were carrying 26 people including 15 U.S. citizens and 11 others, among them British, French, and Turkish military officers as well as Kurdish citizens. All were killed in one of the worst air-to-air friendly fire accidents involving U.S. aircraft in military history.

All the aircraft involved were flying in clear weather with excellent visibility, an AWACS (Airborne Warning and Control System) aircraft was providing surveillance and control for the aircraft in the area, and all the aircraft were equipped with electronic identification and communication equipment (apparently working properly) and flown by decorated and highly experienced pilots.

The hazard being controlled was mistaking a “friendly” (coalition) aircraft for a threat and shooting at it. This hazard, informally called *friendly fire*, was well known and a control structure was established to prevent it. Appropriate constraints were established and enforced at each level, from the Joint Chiefs of Staff down to the aircraft themselves. Understanding why this accident occurred requires understanding why the control structure in place was ineffective in preventing the loss. Preventing future accidents involving the same control flaws requires making appropriate changes to the control structure, including establishing monitoring and feedback loops to detect when the controls are becoming ineffective and the system is migrating toward an accident, that is, moving toward a state of increased risk. The more comprehensive the model and factors identified, the larger the class of accidents that can be prevented.

For this STAMP example, information about the accident and the control structure was obtained from the original accident report [5], a GAO (Government Accountability Office) report on the accident investigation process and results [200], and two books on the shootdown—one originally a Ph.D. dissertation by Scott Snook [190] and one by Joan Piper, the mother of one of the victims [158]. Because of the extensive existing analysis, much of the control structure (shown in Figure 5.3) can be reconstructed from these sources. A large number of acronyms are used in this chapter.



AAI:	Air to Air Interrogation (used with IFF)
ACE	Airborne Command Element: (the commander's representative in the AWACS)
ACO	Airspace Control Order (Guidance for all local air operations in OPC)
AFB	Air Force Base
AI	Airborne Intercept:(a type of radar on fighter aircraft)
ARF	Aircraft Read Files (supplement to the ACO including the ROE)
ASO	Air Surveillance Officer (one of the positions in the AWACS)
ATO	Air Tasking Order (specific mission guidance for the day)
AWACS	Airborne Warning and Control System (a military air traffic control system in the sk
BH	Black Hawk
BSD	Battle Staff Directive (late scheduling changes not making it into the ATO)
CTF	Combined Task Force
CFAC	Combined Forces Air Component(tactical control of all OPC aircraft operating in TAOR and operational control of AF aircraft)
DO	Director of Operations
GAO	U.S. Government Accountability Office
HQ−II	Have Quick (frequency hopping) radios
HUD	Heads Up Display
IFF	Identiification Friend or Foe
JOIC	Joint Operations and Intelligence Center
JSOC	Joint Special Operations Component (search and rescue operations inside Iraq)
JTIDS	Joint Tactical Information Distribution Center (provides ground with picture of airspace occupants)
MCC	Military Coordination Center (operational control of the Black Hawk helicopters)
MD	Mission Director (runs the mission from the ground)
Min Comm	Minimal Communications
NCA	National Command Authority (the President and the Secretary of Defense)
NFZ	No Fly Zone
OPC	Operation Provide Comfort (multi−nation effort to protect Kurdish refugees)
ROE	Rules of Engagement (rules governing actions allowed by the U.S. military forces)
SD	Senior Director (one of the positions in the AWACS)
SITREP	Situation Report
SPINS	Mission−related Special Instructions
TACSAT	Tactical Satellite radios (used by Army helicopter pilots to communicate with MCC
TAOR	Tactical Area of Responsibility (another name for the No−Fly−Zone)
USCINCEUR	U.S. Commander in Chief, Europe
VID	Visual Identification
WD	Weapons Director (a position in the AWACS)

Figure 5.2: The Acronyms Used in this Chapter

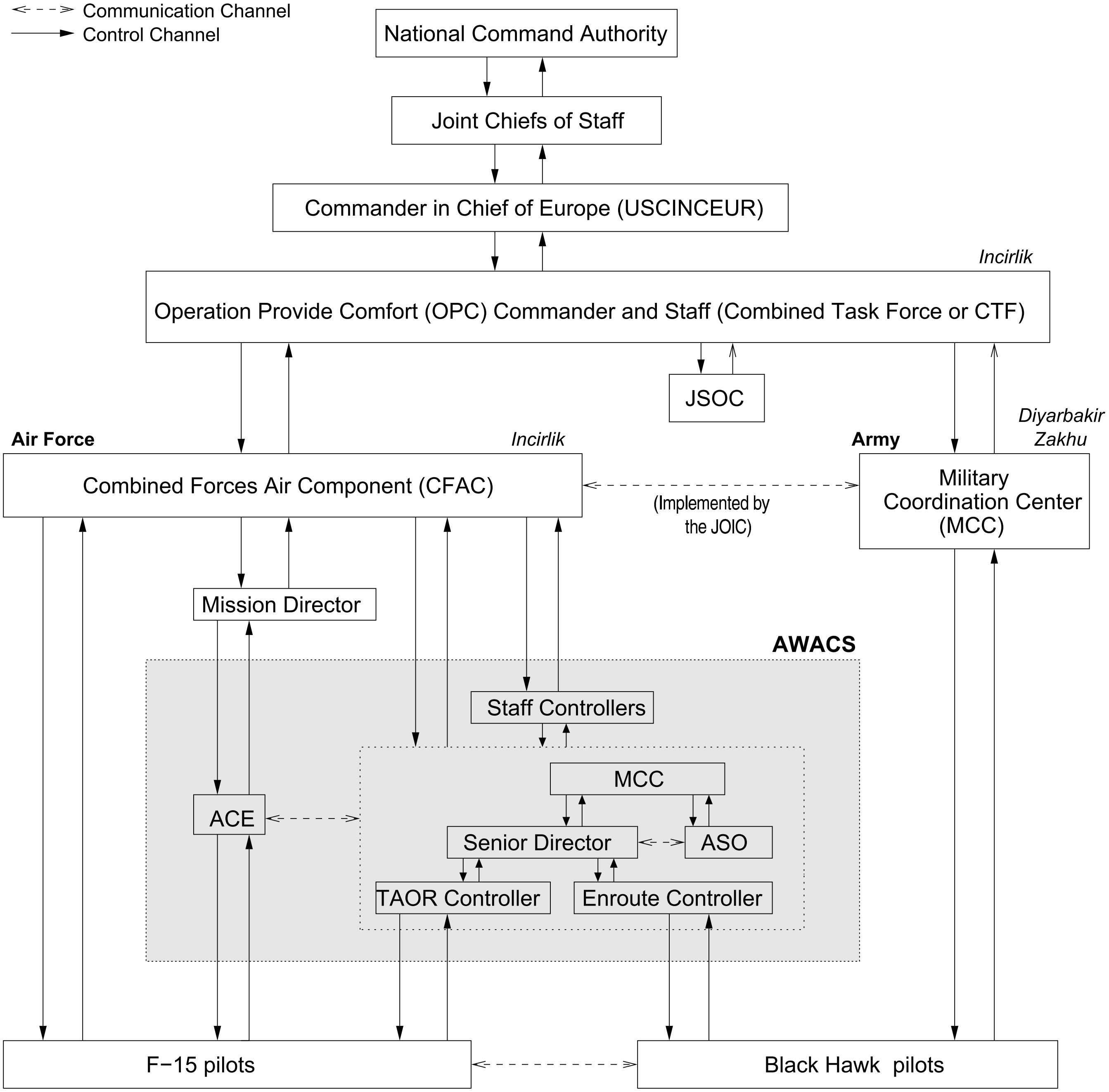

Communication Channel
Control Channel
National Command Authority
Joint Chiefs of Staff
Commander in Chief of Europe (USCINCEUR)
Incirlik
Operation Provide Comfort (OPC) Commander and Staff (Combined Task Force or CTF)
JSOC
Air Force
Incirlik
Combined Forces Air Component (CFAC)
(Implemented by the JOIC)
Army
Diyarbakir
Zakhu
Military Coordination Center (MCC)
Mission Director
AWACS
Staff Controllers
MCC
ACE
Senior Director
ASO
TAOR Controller
Enroute Controller
F−15 pilots
Black Hawk pilots

Figure 5.3: Control Structure in the Iraqi No-Fly Zone.

They are defined in Figure 5.2.

5.2 The Hierarchical Safety Control Structure to Prevent Friendly Fire Accidents

National Command Authority and Commander-in-Chief Europe

When the National Command Authority (the President and Secretary of Defense) directed the military to conduct Operation Provide Comfort (OPC), the U.S. Commander in Chief Europe (USCINCEUR) directed the creation of Combined Task Force (CTF) Provide Comfort.

A series of orders and plans established the general command and control structure of the CTF. These orders and plans also transmitted sufficient authority and guidance to subordinate component commands and operational units so that they could then develop the local procedures that were necessary to bridge the gap between general mission orders and specific subunit operations.

At the top of the control structure, the National Command Authority (the President and Secretary of Defense, who operate through the Joint Chiefs of Staff) provided guidelines for establishing rules of engagement (ROE). ROE govern the actions allowed by U.S. military forces to protect themselves and other personnel and property against attack or hostile incursion and specify a strict sequence of procedures to be followed prior to any coalition aircraft firing its weapons. They are based on legal, political, and military considerations and are intended to provide for adequate self-defense to ensure that military activities are consistent with current national objectives and that appropriate controls are placed on combat activities. Commanders establish ROE for their areas of responsibility that are consistent with the Joint Chiefs of Staff guidelines, modifying them for special operations and for changing conditions.

Because the ROE dictate how hostile aircraft or military threats are treated, they play an important role in any friendly fire accidents. The ROE in force for OPC were the peacetime ROE for the United States European Command with OPC modifications approved by the National Command Authority. These conservative ROE required a strict sequence of procedures to be followed prior to any coalition aircraft firing its weapons. The less aggressive peacetime rules of engagement were used even though the area had been designated a combat zone because of the number of countries involved in the joint task force. The goal of the ROE was to slow down any military confrontation in order to prevent the type of friendly fire accidents that had been common during Operation Desert Storm. Understanding the reasons for the shootdown of the Black Hawk helicopters requires understanding why the ROE did not provide an effective control to prevent friendly fire accidents.

Three System-Level Safety Constraints Related to this Accident:

1. The NCA and UNCINCEUR must establish a command and control structure that provides the ability to prevent friendly fire accidents.

2. The guidelines for ROE generated by the Joint Chiefs of Staff (with tailoring to suit specific operational conditions) must be capable of preventing friendly fire accidents in all types of situations.

3. The European Commander-in-Chief must review and monitor operational plans generated by